UNIT-4

DESIGN

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain i.e. "how to implement?"

⊛. Database Design:

Database design is the organization of data according to a database model. The main purpose of database design is to produce logical and physical designs of the proposed database system to meet the requirement of users and for high performance.

Process: Database modeling and design activities occur in all phases of systems development process. In logical database design we use process called normalization, to build a data model. There are four key steps in logical database modeling and design:

i) Develop a logical data model for each known user interface for the application using normalization principles.
ii) Combine normalized data requirements for all user-interfaces into one logical model, which is con-solidated.
iii) Translate the conceptual E-R data model for the application or enterprise.
iv) Compare the con-solidated logical database design with the translated E-R model and produce one final logical database model for the application.

⊛. Relational Database Model:

The relational database model represents data in the form of related tables, or relations. Each relation (or table) consists of a set of columns called attributes and an arbitrary number of rows called records. Relational data model represents the logical view of how data is stored in the relational databases.

Tuple: Each row in the relation is known as tuple. It contains data records.

Domain: Domain is set of values which is indivisible.

Example:

Columns (Attributes)

Relation (Table)

Stu_id	Stu_name	Stu_address	Dept_id
12	Ramesh	Butwal	5
13	Hari	Kathmandu	3
14	Mahesh	Pokhara	1
15	Sangita	Nepaljung	2

Records or Tuple (Row)

Characteristics/Properties of Relations:

→ Each relation in a database must have a unique name.
→ A relation must not have two attributes with same name. Each attribute must have a distinct name.
→ Duplicate tuples must not be present in relation.
→ Each tuple must have exactly one data value for an attribute.
→ Tuples and attributes of relation do not have to follow a significant order.

⊛ Normalization: [Imp]

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause ~~and~~ insertion, deletion and updation anomalies. It can be considered as a "filtering" or "purification" process to make the design have successively better quality.

⊛ **Normal Forms:** The normal form of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized.

⊛ **De-Normalization:** It is the process of adding redundant data to speed up complex queries involving multiple table JOINS.

⊛ **Functional Dependency:** A functional dependency, denoted by $X \rightarrow Y$, between two sets of attributes X and Y that are subsets of R, such that any two tuples $t1$ and $t2$ in r that have $t1[X] = t2[X]$, they must also have $t1[Y] = t2[Y]$. Functional dependency is represented as FD or f.d.

1) First Normal Form (1NF): A relation is in first normal form if it does not contain any composite or multi-valued attribute. If a relation contain composite or multi-valued attribute, it violates ~~fist~~ first normal form.

Example: Relation STUDENT in table 1 is not in 1NF because of multi-valued attribute STUD_PHONE. Its decomposition into 1NF has been shown in table 2.

STUD_NO	STU_NAME	STU-PHONE	STU_ADDRESS
1.	RAM	9816271721 9817712717	Kathmandu
2.	SURESH	9810162117	Butwal

Table 1

STUD_NO	STU_NAME	STU_PHONE	STU_ADDRESS
1.	RAM	9816271721	Kathmandu
1.	RAM	9817712717	Kathmandu
2.	SURESH	9810162117	Butwal

Table 2

2) Second Normal Form (2NF):

A relation is in second normal form, if a relation is in first normal form and relation must not contain any partial dependency.

Partial dependency ⇒ If proper subset of candidate key determines non-prime attribute.

Some other important concepts:

- Non-prime attribute ⇒ Attributes which are not part of any candidate key.
- Key ⇒ A key is an attribute or set of attributes that uniquely identifies each record in the relation.
- Candidate key ⇒ It is a key whose proper subset is not a key.
- Prime-attribute ⇒ The attributes which are making candidate keys are prime attribute.

Example: Consider the table as below;

STUD_NO	COURSE_NO	COURSE_FEE
1	C1	10000
2	C2	15000
1	C4	20000
4	C3	10000
4	C1	10000
2	C5	20000

Note that there are many courses having the same course fee

5 marks of सोद्दो यो table मात्र बनाउने अनि convert गरको table theory reduce गर्ने

Here,
COURSE_FEE cannot alone decide the value of COURSE_NO or STUD_NO;
COURSE_FEE together with STUD_NO cannot decide the value of COURSE_NO;
COURSE_FEE together with COURSE_NO cannot decide the value of STUD_NO;

Hence,
COURSE_FEE would be a non-prime attribute, as it does not belong to the one only candidate {STUD_NO, COURSE_NO};
But, COURSE_NO → COURSE_FEE i.e, COURSE_FEE is dependent on a COURSE_NO, which is a proper subset of candidate key.
Non-prime attribute COURSE_FEE is dependent on a proper subset of the candidate key, which is a partial dependency and so this relation is not in 2NF.

To convert the above relation to 2NF:
We need to split the table into two tables such as;
Table 1: STUD_NO, COURSE_NO.
Table 2: COURSE_NO, COURSE_FEE

Table 1

STUD_NO	COURSE_NO
1	C1
2	C2
1	C4
4	C3
4	C1
2	C5

Table 2

COURSE_NO	COURSE_FEE
C1	10000
C2	15000
C3	10000
C4	20000
C5	20000

3). Third Normal Form (3NF):

Mostly asked as; explain with example for 5 marks so we can write in this way in short

A relation is in third normal form if it is in second normal form and there is no transitive dependency for non-prime attributes.

Transitive dependency: If $A \rightarrow B$ and $B \rightarrow C$ are two FDs then $A \rightarrow C$ is called transitive dependency.

Example:

Company	City	State	ZIP
Sanima Ltd.	Kathmandu	State3	10169
Shivam Ltd.	Biratnagar	State2	33196
API Company Ltd.	Darchula	State7	21046

write 2NF also in short in exam like this

The above table is not in 3NF. In this example, the city and state are dependent upon ZIP code. To place this table in 3NF, two separate tables would be created: one containing the country name and ZIP code and the other containing city, state, ZIP code pairings.

Company	ZIP
Sanima Ltd.	10169
Shivam Ltd.	33196
API Company Ltd.	21046

City	State	ZIP
Kathmandu	State3	10169
Biratnagar	State2	33196
Darchula	State7	21046.

⊛. Transforming E-R Diagrams into Relations:

Transforming an E-R diagram into normalized relations and then merging all the relations into one final, consolidated set of relations can be accomplished in four steps:

1) Represent entities: Each entity type in the E-R diagram becomes a relation. The identifier of the entity type becomes the primary key of the relation, and other attributes of the entity type become non-primary key attributes of the relation.

2) Represent relationships: Each relationship in an E-R diagram must be represented in the relational database design. How we represent relationship depends on its nature.

iii) Normalize the relations: The relations created in steps i and ii may have unnecessary redundancy. So we need to normalize these relations to make them well structured.

iv) Merge the relations: Across different sets of relations, there may be redundant relations (two or more relations that describe the same entity type) that must be merged and renormalized to remove the redundancy.

⊛. Merging Relations:

Some of the relations may be redundant, we should merge those relations to remove the redundancy.

An example of Merging Relations:

Suppose modeling a user interface or transforming an E-R diagram results in following 3NF relation:

Employee 1 (Emp_ID, Name, Address, Phone).

Modeling a second user interface might result in following relation:

Employee 2 (Emp_ID, Name, Address, Jobcode, Number_of_years)

Because these two relations have same primary key (Emp_ID) and describe the same entity, they should be merged into one relation. The result of merging the relations is the following relation:

Employee (Emp_ID, Name, Address, Phone, Jobcode, Number_of_years).

⊛ Physical file and Database design:

Designing physical files and databases requires certain information that should have been collected during prior SDLC phases. The information includes the following:

→ Normalized relations, including volume estimates
→ Definitions of each attribute.
→ Descriptions of where and when data are used.
→ Requirements for response and data integrity.
→ Descriptions of the technologies used for implementing the files and database so that the range of required decisions and choices for each is known.

⊛. Designing Fields:

A field is the smallest unit of application data recognized by system software, such as a programming language or a database management system. An attribute from a logical database model may be represented by several fields.

A field that can be derived from other database fields is called a calculated field. Some database technologies allow us to explicitly define calculated fields among with other raw data fields. The database technology will either store the calculated value or compute it when requested.

⊛. Designing Physical Tables:

A physical table is a named set of rows and columns that specifies fields in each row of the table. The design of a physical table has two goals: efficient use of secondry storage and data processing speed.

→ The efficierd use of secondry storage relates to how data are loaded on disks. The efficiency of secondry storage depends on factors such as operating system parameters, outside the control of each database.

→ Data are most efficiently processed when they are stored close to one another in secondry memory, thus, minimizing the number of input/output (I/O) operations that must be performed.

#Designing Forms and Reports:

The forms are used to collect or present information on a single item, such as a customer, product or event. Forms can be used for both input and output. Reports on the other hand are used to convey information on a collection of items. Both forms and reports are the product of input and output design. The main difference is that forms provide fields for data input but reports are purely used for reading.

⊛. Process of designing forms and reports: [Imp]

Designing forms and reports is a user-focused activity that typically follows prototyping approach. First we must gain an understanding of the intended user and task objectives by collecting initial requirements. During this process, several questions to "who, what, when, where and how" must be answered.

→ Who will use the form or report?
→ What is the purpose of form or report?
→ When is the form or report needed or used?
→ Where does the form or report need to be delivered and used?
→ How many people need to use or view the form or report?

Gaining and understanding of these questions is a required first step in creation of any form or report. After collecting initial requirements, we structure and refine this information into an initial prototype. Now, we ask users to review and evaluate the prototype. After reviewing the prototype, users may accept the design or request some changes. If changes are needed we will refine it according to user requirements until the design is accepted.

⊛. Deliverables and Outcomes:

i) Narrative overview: It contains overview of characteristics of target users, tasks, system etc. in which form or report will be used.

ii) Sample design: It provides sample design which may be hand drawn using a coding sheet although, in most instances, it is developed using CASE tool.

iii) Testing and usability assessment: This section provides all testing and usability assessment information. Assessing usability depends on speed, accuracy, and satisfaction.

⊛. Formatting forms and reports: [Imp]

1) General Formatting Guidelines: Proper formatting of forms and reports is essential. Following are the general guidelines for the design of forms and reports:

i) **Meaningful titles:** The form or report should contain title that is clear and specific. It should clearly describe the content and use of form or report.

ii) **Meaningful information:** Only the information that is relevant and needed by the user should be displayed on the form or report.

iii) **Balanced layout:** The information should be balanced on the screen or page, i.e, the display should not be too crowded and not to spread out.

iv) **Easy Navigation:** It should be possible for the user to easily move forward and backward through the contents of form or report.

2) Guidlines for displaying contents: The way the form or a report appears to the human eye has a lot of impact on the user so, we follow some guidlines for better display of contents:

i) **Highlighting information:** Highlighting the information will enhance the appearance of the output. Highlighting of information can be carried out using different methods such as colour difference, intensity difference, underlining, font and size differences etc.

ii) **Using Colour:** Use of appropriate colours while designing has several advantages like strikes the eye, draws attention to warnings, use of colours in graphs and charts helps in better understanding etc.

iii) **Displaying text:** We should use appropriate punctuation wherever required. The text should be properly spaced and there should be blank line between paragraphs.

iv) **Designing tables and lists:** We should use meaningful labels to all columns and rows and seperate labels from other information by using highlighting. Displayed data should be sorted in meaningful order.

⊛ Paper based vs. Electronic reports:

Paper based reports	Electronic reports
i) Paper based reports are stored locally in filing cabinets.	i) Electronic reports are stored in file systems of computer.
ii) Editing, and Copying is difficult in paper based reports.	ii) Editing, and Copying is easier in electronic reports.
iii) For multiple users to access one needs a set of reports per each accessing person.	iii) Multiple users may access single report simultaneously.
iv) It can be delivered by mail or manually.	iv) It can be delivered by networks, disks, flash memory etc.
v) Paper reports take time for searching.	v) Electronics reports are faster to search.

⊛ Assessing Usability:

(less imp)

Usability typically refers to the following three characteristics:

i) Speed: Can you complete a task efficiently?

ii) Accuracy: Does the system provide what you expect?

iii) Satisfaction: Do you like using the system?

General design Guidelines for usability of forms and reports:

Usability Factor	Guideline for achievement of usability
Consistency	→ Consistent use of terminology, formatting titles and navigation within and across outputs.
Organization	→ Text and data should be aligned and sorted for efficient navigation and entry.
Clarity	→ Outputs should be clear to the user and units of measure should be clearly indicated.
Format	→ Information format should be consistent between entry and display.
Flexibility	→ Information should be viewed and retrieved in a manner most convenient to the user.

#Designing Interfaces and Dialogues:

Interface design focuses on how information is provided to and captured from users. Dialogue design focuses on the sequencing of interface displays. The design of interfaces and dialogues is the process of defining the manner in which humans and computers exchange information.

⊛. Measures of Usability:

i) Learnability: How difficult is it for a user to perform task for the first time?

ii) Efficiency: How quickly can users perform tasks?

iii) Error rate: How many errors might a user encounter, and how easy it is to recover from those errors?

iv) Memorability: How easy is it to remember how to accomplish a task?

v) Satisfaction: How enjoyable is the system to use?

⊛. Process of Designing Interfaces and dialogues:

Similar to designing forms and reports, the process of designing interfaces and dialogues is a user-focused activity. This means we should follow a prototyping methodology iteratively collecting information, and making refinements.

⊛. Deliverables and Outcomes:

Similar to that of forms and reports.

i) Narrative overview

ii) Sample design

iii) Testing and usability assessment.

⊛. Interaction methods and devices:

Methods of Interacting: Methods of interacting includes following:

i) Command language interaction: This type of interaction requires users to remember command syntax and semantics. The user enters explicit statements to invoke operations within system.

ii) Menu interaction: A menu is simply a list of options. When an option is selected by the user, a specific command is invoked. Menus are the most widely used interfaces for interaction.

iii) Form interaction: It allows users to fill in the blanks when working with a system. Form interaction is important and effective for both the input and presentation.

iv) Object-based interaction: Using icons is the most common method for implementing object-based interaction. Icons are the graphic symbols. Icons take little space and can be quickly understood by users.

v) Natural language interaction: Natural language interaction is being applied with both keyboard and voice entry systems.

⊛. Designing Interfaces:

User interface is the front-end application view to which user interacts in order to use the software. The software becomes more popular if its user interface is attractive, simple to use, responsive etc. It consists of following:

1) Designing Layouts: For designing layouts standard formats similar to paper-based forms and reports should be used. Screen navigation on data entry screens should be left-to-right, top-to-bottom as on paper forms. Flexibility and consistency are primary design goals.

2) Structuring data entry:

→ Never require data that are already on-line or that can be computed.
→ Always provide default values when appropriate.
→ Make clear the type of data units requested for entry.
→ Use character replacement when appropriate.

3) Controlling data input:

→ One objective of interface design is to reduce data entry errors.
→ Role of systems analyst is to anticipate user errors and design features into the system's interfaces to avoid, detect and correct data entry mistakes.

4) Providing Feedback:

→ Interface design should provide status information as feedback like keeping users informed of what is going on the system.
→ It should provide error and warning messages.

5) Providing Help:

→ We should try to place ourself in user's place when designing help.
→ Help messages should be short and to the point.

⊛. Designing Dialogues:

The process of designing the overall sequences that users follow to interact with an information system is called dialogue design. The dialogue design process consists of three major steps:

→ Designing the dialogue sequence
→ Building a prototype
→ Assessing usability.

In designing the dialogue sequence, we must first gain the understanding of how users might interact with the system. Then we build the prototype of the system meeting user's requirements. For a dialogue to have high usability, it must be consistent in form, function and style.

⊛. Designing Interfaces and Dialogues in Graphical Environments:

Graphical user interface (GUI) environments have become the de facto for human-computer interaction. When designing graphical interfaces we encounter different issues, so numerous factors must be considered. Some factors are common to all GUI environments, whereas others are specific to the single environment. In most discussions of GUI programming, two rules repeatedly emerge as composing the first step to become an effective GUI designer:

→ Become an expert of the GUI environment.
→ Understand the available resources and how they can be used.