



AIQ 3211 DATABASE CONSTRUCTION AND MANAGEMENT

Mr. Jackson Alunga

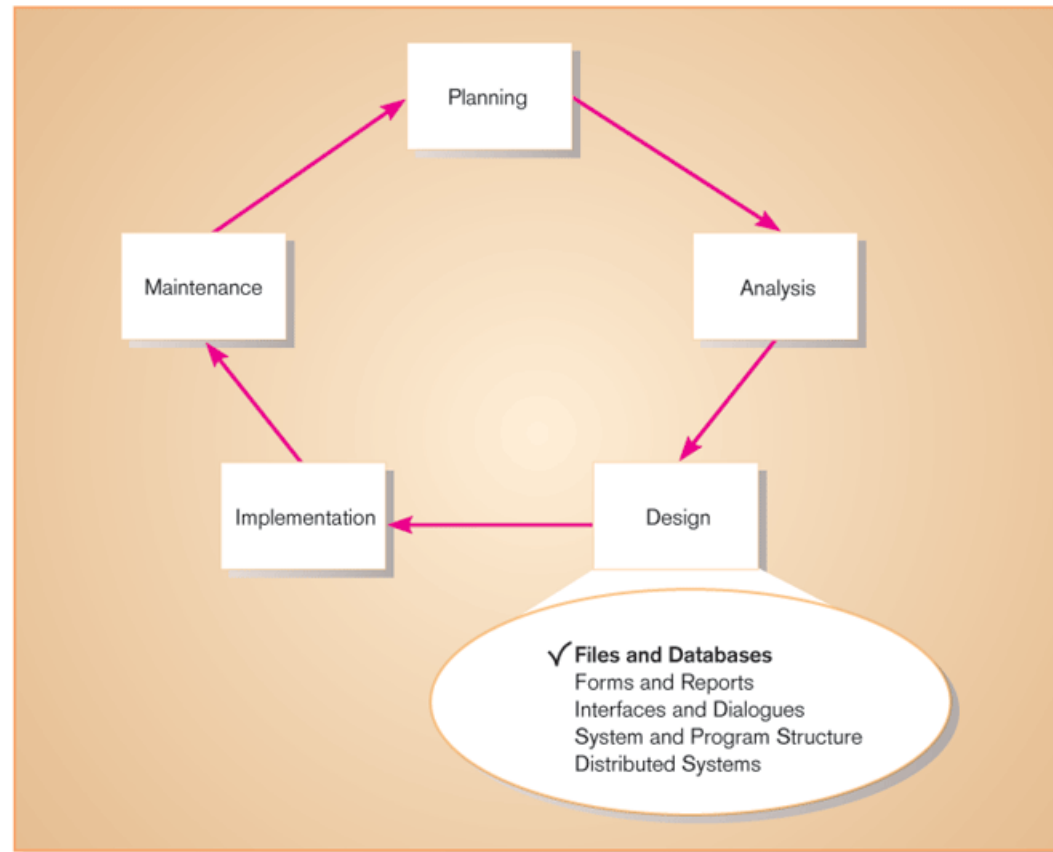
jalunga@tukenya.ac.ke

Learning Objectives

- ✓ Concisely define each of the following key database design terms: relation, primary key, normalization, functional dependency, foreign key, referential integrity, file organization, and index.
- ✓ Explain the role of designing databases in the analysis and design of an information system.
- ✓ Explain when to use different types of file organizations to store computer files.
- ✓ Describe the purpose of indexes and the important considerations in selecting attributes to be indexed.

Introduction

Figure 10-1 Systems development life cycle with design phase highlighted



Database Design

- File and database design occurs in two steps.
- Develop a logical database model, which describes data using notation that corresponds to a data organization used by a database management system.
 - Relational database model.
- Prescribe the technical specifications for computer files and databases in which to store the data.
 - Physical database design provides specifications.
- Logical and physical database design in parallel with other system design steps.

The Process of Database Design (Cont.)

- Four key steps in logical database modeling and design:
 - Develop a logical data model for each known user interface for the application using normalization principles.
 - Combine normalized data requirements from all user interfaces into one consolidated logical database model (view integration).
 - Translate the conceptual E-R data model for the application into normalized data requirements.
 - Compare the consolidated logical database design with the translated E-R model and produce one final logical database model for the application.

Physical Database Design

- Key physical database design decisions include:
 - Choosing storage format for each attribute from the logical database model.
 - Grouping attributes from the logical database model into physical records.
 - Arranging related records in secondary memory (hard disks and magnetic tapes) so that records can be stored, retrieved and updated rapidly.
 - Selecting media and structures for storing data to make access more efficient.

Table 11.2 Common Oracle 9i Datatypes

Data Type	Description
Varchar2	Stores variable length character data. Maximum length is 4,000 bytes.
Char	Stores fixed length character data. If the value has fewer characters than specified, the remaining characters are padded with blanks. Maximum length is 2,000 bytes.
Number (precision, scale)	Stores fixed or floating-point numbers. Precision is the total number of digits. Scale determines where rounding occurs. Maximum precision is 38.
Date	Date values ranging from 4712 B.C. to 9999 A.D.
Timestamp	Date values with precision down to a fraction of a second with up to nine places after the decimal.
Long	Stores variable length character data, is like VARCHAR2, but maximum length is 2 GB.
Lob	There are four types: Binary LOB (BLOB), Character LOB (CLOB), National Character LOB (NCLOB), and Binary file (BFILE). The data type is used to store large objects such as images, video files, and character data in a number of languages.

Deliverables and Outcomes

- Logical database design
 - Must account for every data element on a system input or output.
 - Normalized relations are the primary deliverable.
- Physical database design
 - Convert relations into database tables.
 - Programmers and database analysts code the definitions of the database.
 - Written in Structured Query Language (SQL).

Relational Database Model

- **Relational database model:** data represented as a set of related tables or relations.
- **Relation:** a named, two-dimensional table of data. Each relation consists of a set of named columns and an arbitrary number of unnamed rows.

Relational Database Model (Cont.)

- Relations have several properties that distinguish them from nonrelational tables:
 - Entries in cells are simple.
 - Entries in columns are from the same set of values.
 - Each row is unique.
 - The sequence of columns can be interchanged without changing the meaning or use of the relation.
 - The rows may be interchanged or stored in any sequence.

Well-Structured Relation and Primary Keys

➤ **Well-Structured Relation (or table)**

- A relation that contains a minimum amount of redundancy;
- Allows users to insert, modify, and delete the rows without errors or inconsistencies.

➤ **Primary Key**

- An attribute whose value is unique across all occurrences of a relation.
- All relations have a primary key.
- This is how rows are ensured to be unique.
- A primary key may involve a single attribute or be composed of multiple attributes.

Normalization and Rules of Normalization

- **Normalization:** the process of converting complex data structures into simple, stable data structures.
- First Normal Form (1NF)
 - Unique rows, no multivalued attributes.
 - All relations are in 1NF.
- Second Normal Form (2NF)
 - Each nonprimary key attribute is identified by the whole key (called full functional dependency).

Rules of Normalization (Cont.)

- Third Normal Form (3NF)
 - Nonprimary key attributes do not depend on each other (i.e. no transitive dependencies).
- The result of normalization is that every nonprimary key attribute depends upon the whole primary key.

Functional Dependencies and Primary Keys

➤ Functional Dependency

- A particular relationship between two attributes.
- For a given relation, attribute B is functionally dependent on attribute A if, for every valid value of A, that value of A uniquely determines the value of B.
- The functional dependence of B on A is represented by $A \rightarrow B$.

Functional Dependencies and Primary Keys (Cont.)

- Functional dependency is not a mathematical dependency.
- Instances (or sample data) in a relation do not prove the existence of a functional dependency.
- Knowledge of problem domain is most reliable method for identifying functional dependency.

Transforming E-R Diagrams into Relations

- It is useful to transform the conceptual data model into a set of normalized relations.
- Steps
 - *Represent entities.*
 - *Represent relationships.*
 - *Normalize the relations.*
 - *Merge the relations.*

Representing Entities

- Each regular entity is transformed into a relation.
- The identifier of the entity type becomes the primary key of the corresponding relation.

Representing Entities

- The primary key must satisfy the following two conditions.
 - The value of the key must uniquely identify every row in the relation.
 - The key should be nonredundant.
- The entity type label is translated into a relation name.

Binary 1:N and 1:1 Relationships

- The procedure for representing relationships depends on both the degree of the relationship – unary, binary, ternary – and the cardinalities of the relationship.
- **Binary 1:N Relationship:** is represented by adding the primary key attribute (or attributes) of the entity on the one side of the relationship as a foreign key in the relation that is on the many side of the relationship.

Binary 1:N and 1:1 Relationships (Cont.)

- **Binary or Unary 1:1 Relationship:** represented by any of the following choices:
 - Add the primary key of A as a foreign key of B.
 - Add the primary key of B as a foreign key of A.
 - Both of the above.

Binary and Higher-Degree M:N Relationships (Cont.)

- **Binary and Higher-Degree M:N relationships**
 - Create another relation and include primary keys of all relations as primary key of new relation.

Unary Relationships

► **Unary 1:N Relationship**

- Is modeled as a relation.
- Primary key of that relation is the same as for the entity type.
- Foreign key is added to the relation that references the primary key values.

- **Recursive foreign key:** A foreign key in a relation that references the primary key values of that same relation.

Unary Relationships

- **Unary M:N Relationship**
- Is modeled as one relation.
- Create a separate relation the represent the M:N relationship.
- Primary key of new relation is a composite key of two attributes that both take their values from the same primary key.
- Any attribute associated with the relationship is included as a nonkey attribute in this new relation.

Merging Relations

- Purpose is to remove redundant relations.
- The last step in logical database design.
- Prior to physical file and database design.

View Integration Problems

- Must understand the meaning of the data and be prepared to resolve any problems that arise in the process.
- **Synonyms:** two different names used for the same attribute.
 - When merging, get agreement from users on a single, standard name.

View Integration Problems (Cont.)

- **Homonyms:** a single attribute name that is used for two or more different attributes.
 - Resolved by creating a new name.
- **Dependencies between nonkeys:** dependencies may be created as a result of view integration.
 - In order to resolve, the new relation must be normalized.

View Integration Problems (Cont.)

- **Class/Subclass:** relationship may be hidden in user views or relations.
 - Resolved by creating a new name.

File Organizations

- **File organization:** a technique for physically arranging the records of a file.
- **Physical file:** a named set of table rows stored in a contiguous section of secondary memory.

File Organizations (Cont.)

- **Sequential file organization:** a file organization in which rows in a file are stored in sequence according to a primary key value.
- **Hashed file organization:** a file organization in which the address for each row is determined using an algorithm.
- **Pointer:** a field of data that can be used to locate a related field or row of data.

Arranging Table Rows (Cont.)

- Objectives for choosing file organization
 - Fast data retrieval.
 - High throughput for processing transactions.
 - Efficient use of storage space.
 - Protection from failures or data loss.
 - Minimizing need for reorganization.
 - Accommodating growth.
 - Security from unauthorized use.
 - Protection from failures or data loss.
 - Minimizing need for reorganization.
 - Accommodating growth.
 - Security from unauthorized use.

Indexed File Organization

- **Indexed file organization:** a file organization in which rows are stored either sequentially or nonsequentially, and an index is created that allows software to locate individual rows.
- **Index:** a table used to determine the location of rows in a file that satisfy some condition.
- **Secondary keys:** one or a combination of fields for which more than one row may have the same combination of values.

Indexed File Organization (Cont.)

- Main disadvantages are:
 - Extra space required to store the indexes; and
 - Extra time necessary to access and maintain indexes.
- Main advantages are:
 - Allows for both random and sequential processing.
- Guidelines for choosing indexes:
 - Specify a unique index for the primary key of each table.
 - Specify an index for foreign keys.
 - Specify an index for nonkey fields that are referenced in qualification, sorting and grouping commands for the purpose of retrieving data.

Summary

- ✓ Concisely define each of the following key database design terms: relation, primary key, normalization, functional dependency, foreign key, file organization, and index.
- ✓ Explain the role of designing databases in the analysis and design of an information system.
- ✓ Explain when to use different types of file organizations to store computer files.
- ✓ Describe the purpose of indexes and the important considerations in selecting attributes to be indexed.