INTRODUCTION TO DBMS

Database is an organized collection of information. To manage databases, you need database management systems. DBMS or database management system is a software package for defining and managing a database. It mainly stores or retrieves the data from the database upon request.

Types of databases

There are five main types of databases:

- Flat
- Network
- Hierarchical
- Relational
- Object-oriented

Flat database. is a relatively simple database system, in which each database is contained in a single table. Flat databases are good for small applications. It cannot be linked to data in other files.

Network database uses set theory to provide a tree-like hierarchy. The child table can have more than one parent. Therefore, network model can support many-to-many relationships. This model had its origins in the Conference on Data Systems Languages (CODASYL).

Hierarchical database model is a subset of network model. It defines hierarchically arranged data. In this, a single table acts as the root of the database from which other tables branch out. A parent may have more than one child but a child can have only one parent. To get to a low-level table, you start at the root and work your way down through the tree. One problem with this system is that the user must know how the tree is structured in order to find anything.

Relational database model is an information system that presents information as rows contained in a collection of tables, each table possessing a set of one or more columns.

Object-oriented database model takes object-oriented features and incorporates these into DBMS. It provides the semantics of C++, Java etc. to provide full-featured database programming capability. Instead of storing raw data, here objects are stored. It helps in the unification of database and application development, thus requiring less code. This one-to-one mapping of object programming language objects to database objects has two benefits: it provides higher performance management of objects, and it enables better management of the complex interrelationships between objects.

Oracle 8i is an ORDBMS (Object Relational Database Management System). It provides an engine, which includes object oriented programming model.

Introduction to RDBMS

Relational database model is an information system that presents information as rows contained in a collection of tables, each table possessing a set of one or more columns. Thus, relational database uses two-dimensional tables to store data or information. Dr. Codd gave the relational model for database systems. According to him, these are the three main characteristics of RDBMS:

- Collection of objects or relations to store the data
- A set of operators that can act on relations to produce other relations
- Data integrity for accuracy

Relational database can have one or many tables.

Relational Database theory

- Every entity has a set of attributes that describe the entity. For example, an entity named Course would describe the courses offered at a college. An entity is implemented as a database table.
- A row is a single instance of attribute values. For example, a row in the Course entity would describe a single course offered by the college.
- Some of an entity's attributes uniquely identify each row in that entity. This set of attributes is called the primary key. For example, Student_ID is the primary key because it uniquely defines each student. Every table must have a primary key, which enforces data integrity.
- None of the attributes that contain the primary key can be null.
- A foreign key is an attribute in one entity, whose values must exist as the primary key in another entity. Referential integrity is achieved, when the set of values in a foreign key column is restricted to the primary key that it references or to the null value.
- Entities are related to one another.
- The order of the rows in an entity is arbitrary.
- The order of the attributes in an entity is arbitrary.

A relational database is a collection of two-dimensional tables. **Table** is a storage structure in RDBMS. Each table has rows and columns. A single row represents data for that particular record. Each column has a particular field. The column has a data type (string, integer, BLOB etc.) and possibly, a default value. The column can also have a NOT NULL requirement, which means that you can't add a row that contains no value in this column or the column must have a unique value.

A table's **primary key** is a column that uniquely identifies each row in the table. Primary key column cannot have null values.

There can be a **foreign key** in a table, which determines how any two tables relate to each other. In a table, a foreign key is a column or set of columns, whose values are restricted to those of the primary key in another table. Primary and foreign keys work together to enforce referential integrity. Referential integrity defines what action the server should take, when it updates or deletes a row and there are related rows in other tables.

Here is an employee table called `employees`, which has:

- ***fields*** like `employee name, employee ID, department number`
- ***record*** is the whole row like `Susan Miller, 1234, 300`
- ***primary key*** is `employee ID`
- ***foreign key*** is `department number`, which is the primary key in `dept` table

Employee Name	**Employee ID**	**Department Number**
Susan Miller	1234	300
James Young	5678	100
Ira Yu	1521	200
Joe Ford	8521	100

RDBMS can be modified and accessed by **`SQL`** (structured query language). `SQL` is `ANSI` standard non-procedural language for operating upon relational databases. `SQL` plus is Oracle proprietary interface for executing `SQL` statements. `SQL` plus statements can be abbreviated and does not need termination characters (unlike SQL). **`PL/SQL`** (procedural language/`SQL`) is procedural language extension to `SQL`. `PL/SQL` is used by Oracle Server and other Oracle tools.

In order to have an optimal database design, normalization theory can be applied.

Normalization theory involves relations (tables), attributes (columns) and dependency of attributes upon one another. With normalization, you can minimize redundant and inconsistent data and avoid update anomalies. Using normalization rules, all columns should contain a single piece of information and should depend only on primary key. There are five normal forms, with the first being the most generally applicable.

First normal form

First normal form (1NF) says that arrays or other repeating fields should not be used. For example, if you create a `product order table` with fields – `ProductID1, ProductPrice1, ProductID2, ProductPrice2` etc. This would not be an optimized design. Instead, you should have an Item table with the primary key of `Orders` table as its foreign key.

Second normal form

For a table to qualify for second normal form, it should be in first normal form (1NF) and all of the data in the table must be dependent on the value of the primary key. It avoids unnecessary duplication of data. For example, if you have a table called `Articles`, where you have all the articles published by an author with the author information. This would not be a very optimized design because if you have one author with many articles, you are repeating the author's information for those rows. Instead, you can have an `Articles` table and an `Author` table with a link between them using `AuthorID`.

Third normal form

For a table to qualify for third normal form, it must be in 2NF, 1NF and each of columns in that table except those used as keys, must not be interdependent. You don't want a table to contain information already defined somewhere else or which can be calculated from existing values.

Commonly used relational database management systems are made by Oracle, Sybase, Informix, IBM, Microsoft. Oracle provides multi-platform support.

Designing Relational Databases

The **first step** in designing a database application is **gathering requirements**. This would include functional requirements, data requirements, performance requirements.

The **second step** is developing a logical data model. A logical data model is representation of the data elements used by an enterprise and relationships between those data elements. One of the most common methods for developing a logical model is entity relationship modeling. An entity is an object, with a defined set of attributes. There would be certain relationships between different entities. These relationships will enforce business rules.

Entity Relationship Model

Data is divided into discrete categories or entities. An entity relationship model (ER model) is an illustration of various entities in the business and the relationships between them. An ER model is built during the analysis phase of the development life cycle. ER models separate the information required by a business from the activities performed within a business. Although businesses can change their activities, the type of information tends to remain constant. Therefore, the data structures tend to remain constant.

Components of ER model

ER model has entity, attribute, relationship.

Entity is an object, for which information needs to be known. For example, in an office, this would include employees, departments, inventory, orders, products etc.

Attribute describes or qualifies an attribute. For example, each employee has an employee ID, hire date, department number, job title, salary. Attributes may be optional or mandatory.

Relationship is a named association between entities. For example, employees and departments have a specific relationship between them.

Advantages of ER model

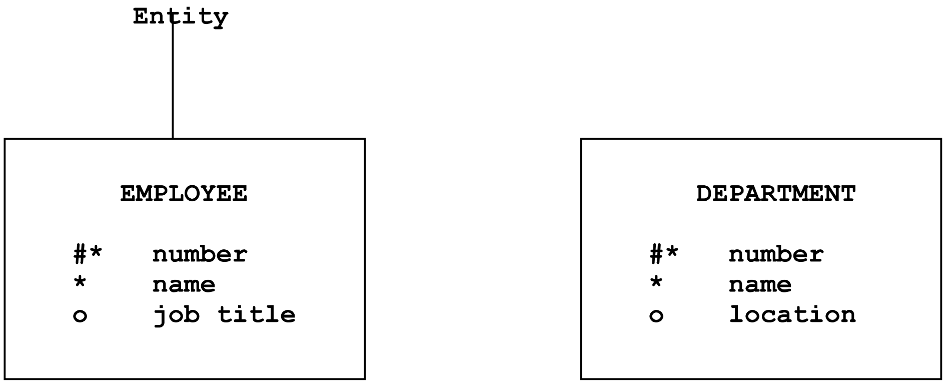
- Information is organized in a clear and precise format
- It provides a clear scope of the information requirement
- Provides an easily understood pictorial map for the database design

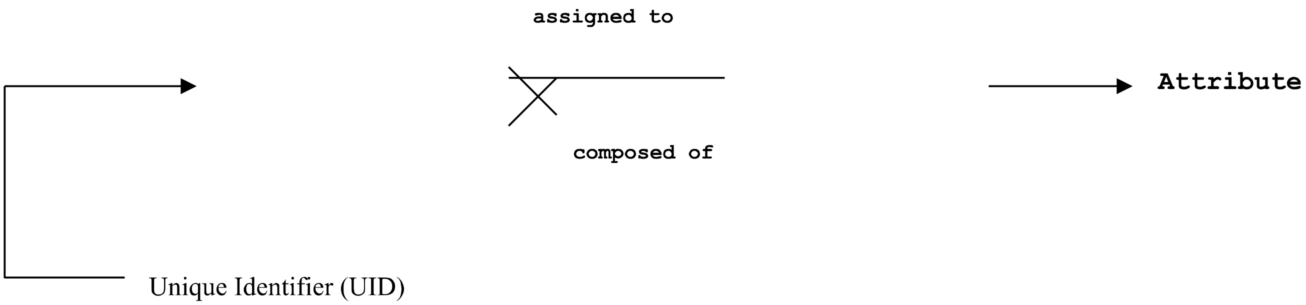
ER modeling conventions

Entity	**Attribute**	**Relationships**
Soft box with any dimensions		Dashed line used for an optional element (“may be”)
Singular, unique entity name	Singular names	Solid line used for mandatory element (“must be”)
Entity name in upper case	Attribute name in lower case	Crow’s foot used for degree element (“one or more”)
Optional synonymn names in uppercase with parentheses	Mandatory attributes are tagged with an asterik (*) and optional attributes with an “o”	Single line used for degree element indicating “one and only one”

We need to discuss little bit more about relationships. Each direction of the relationship contains:

A *name*
An *optionality* (*must be* or *may be*)
A *degree* (*one and only one* or *one or more*)





Unique Identifier (UID)

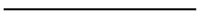
Primary marked with “#” and secondary with “(#)”

Relationships

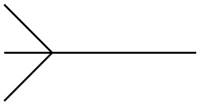
There can be three different types of relationships:

- One-To-Many relationship links a row of data in one table to multiple rows of data in the other table.
- One-To-One relationship links a row of data from one table to only one row in the other table.
- Many-To-Many relationship links many rows in one table to many rows in the other table.

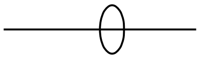
Relationships can be represented like this:



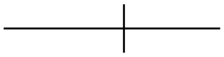
The one side of the relationship



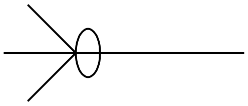
The many side of the relationship



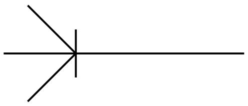
Zero or one row may exist



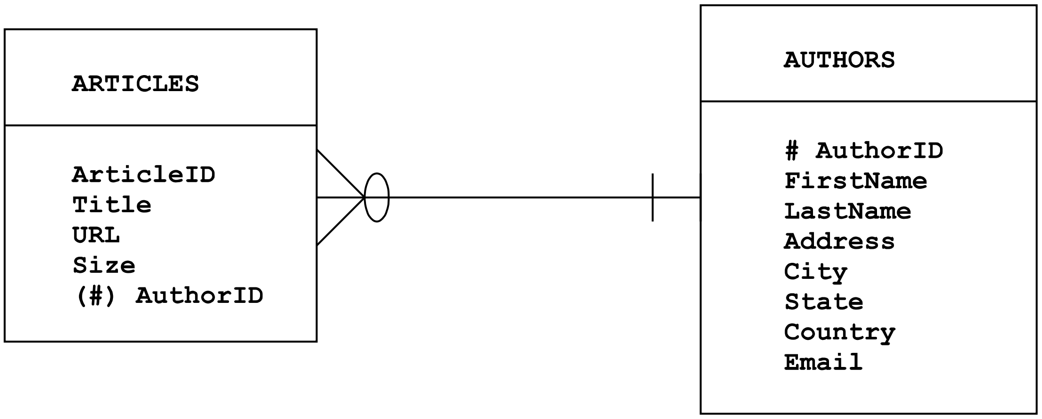
Exactly one row must exist



Zero to many rows may exist



At least one row must exist and many may exist



Defining the relationship between the two tables

Here, you see that each author can have multiple articles and each article has one author. `AuthorID` is a primary key in the `authors` table, but a foreign key in the `articles` table.

The third step is **implementing the logical data model**. You can do that by using different DDL's defined later in the other session. You can modify or manipulate the data using DML statements.