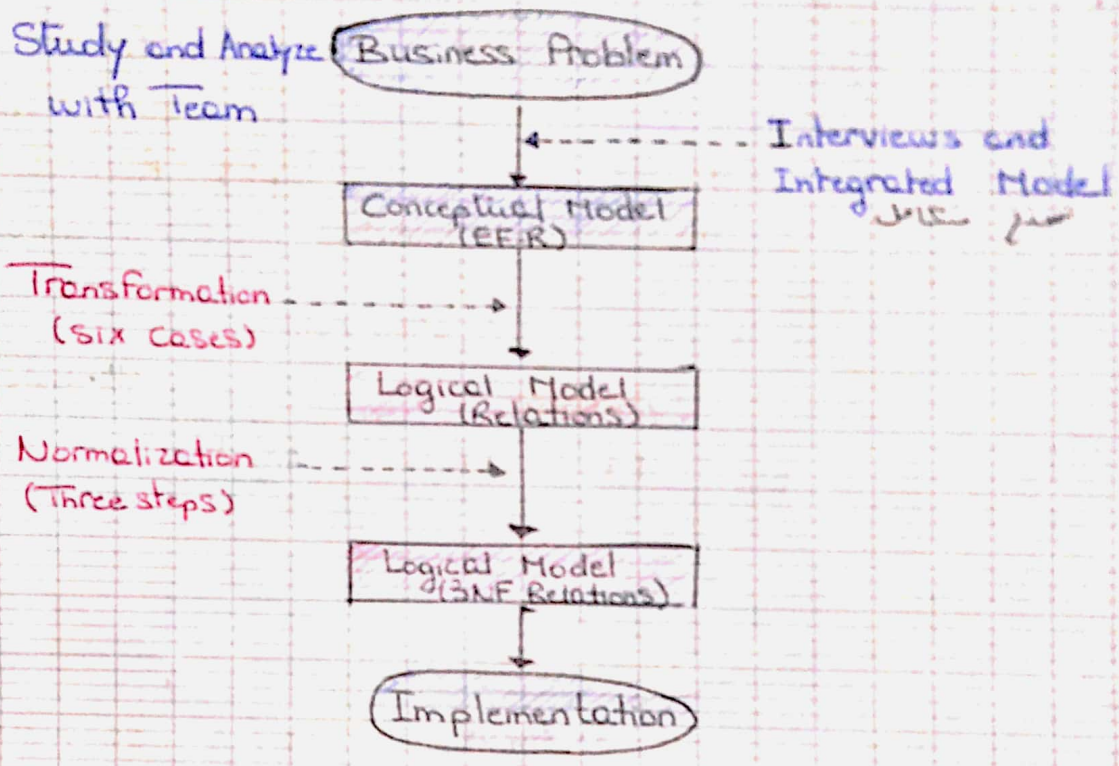
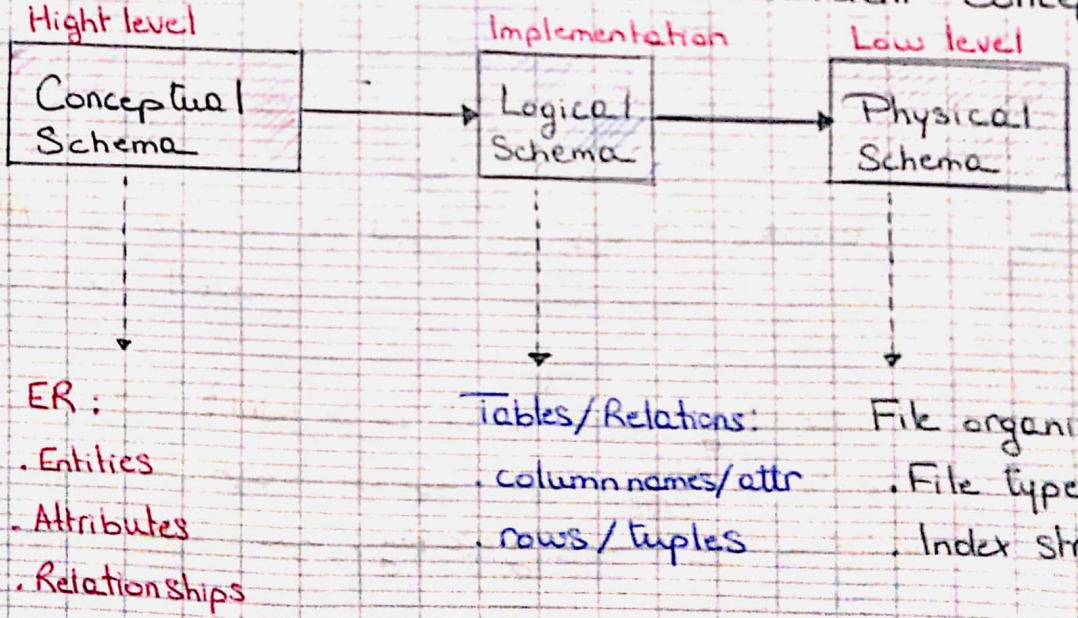
CH4: The Relational Data Model

→ Steps in Database Problem Solving



→ Different Schemas are Based on Different Concepts



↳ Formal Definition - Schema

- The Schema (or description) of a Relation:
  - Denoted by $R(A1, A_2, \ldots A_n)$
  - R is the name of the relation
  - The attribute of the relation are $A_1, A_2 \ldots A_n$

Ex:

Customer (Cust_id, Cust_name, Address)

- Customer is the relation name
- Define over the 3 attributes:
  Cust_id; Cust_name; Address.

- Each Attribute has a domain or a set of valid values. (قيمة معطاة لـ attr معيّن)
  For example, the domain of cust_id is "number"

- The domain is defined by a Type
  A type is the class of atomic values:
  - integers, reals, strings
  - integers between 15 and 80
  - String of (up to) 50 character

  For example the domain of cust_id have a type:
  - Number : 6 digit integer.

Example

Domain Declaration:

Name = String (30), DollarPrice = Decimal (10,2)

Relation Schema:

Product (Prodname: Name , Price : DollarPrice ,
Category: Name , Manufacturer : Name)

Instance

Prodname	Price	Category	Manufacturer
gizmo	19,99	gadgets	Gizmo works
Power gizmo	29,99	gadgets	Gizmoworks
Single Touch	149,99	Photography	Canon
MultiTouch	203,99	Household	Hitachi

→ Relational Integrity Constraints

- Constraints are conditions that must hold on all valid relations states

- There are three main types of constraints:
  - Key constraints
  - Entity integrity constraints
  - Referential integrity constraints

- Another implicit constraints is the domain constraint.
  Every value in a tuple must be from the domain of its attribute (or it could be null if allowed for that attribute)

1. Key Constraints

- SuperKey of a relation (composite Key)
  Is a set of attributes with the following condition:
  - No 2 tuples of any valid relation state (r(R)) will have the same superKey

- Primary Key of a relation
  - If a relation has several candidate Keys one is choosen to be a primary Key
  - The primary Key attributes are underlined
  - The primary Key value is used to uniquely identify each tuple in a relation
  - The primary Key can't be NULL
- Foreign Keys

2. Entity integrity constraints

- The primary Key attributes of each relation schema in the database schema cannot have null values in any tuple

- If the primary Key is composite (several attributes), NULL is not allowed in any of these attributes.

- Note: Other attribute of R may be constrained to disallow null values, even though they are not members of the primary Key.

3. Referential Integrity Constraint

- A constraint involving two relations
- Used to specify a relationship among tuples in two relations (the referencing relation and the referenced relation)

- Tuples in the referencing relation (جدول) have attributes called foreign key that reference the primary key of the referenced relation (جدول)

- Each foreign key value must match a primary key value or must be null

- Delete Rules:
  - Restrict: Don't allow delete of "parent" side if related rows exist in "dependent" side
  - cascade: Automatically delete "dependent" side rows that correspond with the "parent" side row deleted
  - Set-to-null: The foreign key in the dependent side set to null if deleting from the "parent" side → not allowed for weak entities

Example of a relationnal model

Customer

CustID	Custname	custaddress	Custcity	Custstate	CustPostalCode

Order

OrderID	Orderdate	CustID

Order Line

OrderID	ProductID	Ordered Quantity

Product

ProductID	ProductDescription	Product Finish	ProductPrice

```
CREATE TABLE Customer_T
      (CustID              NUMBER(11,0)     NOT NULL,
       CustName            VARCHAR2(25)     NOT NULL,
       CustAddress         VARCHAR2(30)
       Cust City           VARCHAR2(20)
       Cust state          CHAR(2)
       Cust Postal Code    VARCHAR2(9)
CONSTRAINT customer_PK PRIMARY KEY (CustomerID));

CREAT TABLE Order_T
      (OrderID             NUMBER(11,0)     NOTNULL,
       OrderDate           DATE DEFAULT SYSDATE
       CustomerID          NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (orderID),
CONSTRAINT Order_FK FOREIGN KEY (CustID) REFERENCES
                                  customerT(CustID));
```