CLASSICAL APPROACHES

- **HIERARCHICAL:**

  data represented in a tree structure.

  DISADVANTAGE:
  limited structuring capabilities, low level data manipulation, systems require mainframe.

- **NETWORK:**

  data represented in a network structure.

  ADVANTAGE:
  more structuring capabilities.

  DISADVANTAGE:
  complex data manipulation by navigation, systems require mainframe.

- **RELATIONAL:**

RELATIONAL APPROACH

DATA STRUCTURE
Example: supplier - part - supply.

supplier

S#	SNAME	LOCATION
S1	JOHN	LONDON
S2	LEON	PARIS

supply

S#	P#	QTY
S1	P2	120
S1	P3	20

part

P#	PNAME
P2	SCREW

PROPERTIES:

- TABLES CONTAIN ATOMIC VALUES.
- RELATIONSHIPS EXPRESSED BY PRIMARY AND FOREIGN KEYS.

INTEGRITY:

- DEPENDING ON DBMS (not standard) (referential integrity does not hold in the example above).

RELATIONAL APPROACH

DATA MANIPULATION:

- **RELATIONAL ALGEBRA**
- **RELATIONAL CALCULUS**
- **SQL: STRUCTURED QUERY LANGUAGE**

STRUCTURE AND OPERATIONS ARE MORE OR LESS INDEPENDENT OF EACH OTHER.

EXAMPLE INHERENT (STRUCTURAL) PROPERTY: REFERENTIAL INTEGRITY IS NOT USED IN RELATIONAL DML.

USERS:

- **APPLICATION PROGRAMMER:**
  MUST BE AWARE OF LACK OF STRUCTURAL SUPPORT.
- **END USER:**
  CANNOT BE ASSUMED TO BE AWARE OF THE LACK, SO ONLY SELECTION ADMITTED.

RELATIONAL APPROACH
relational algebra

OPERATIONS:

FROM GENERAL SET THEORY:

- **UNION**
- **INTERSECTION**
- **DIFFERENCE**

FROM RELATIONAL THEORY:

- **CARTESIAN PRODUCT**
- **JOIN**
- **RESTRICTION**
- **DIVIDE**

OPERATIONS EXISTED EVEN BEFORE THE CONCEPT OF A DATABASE WAS INVENTED !!

RELATIONAL APPROACH
data structure

supplier

S#	SNAME	LOCATION
S1	JOHN	LONDON
S2	LEON	PARIS

supply

S#	P#	QTY
S1	P2	120
S1	P3	20

part

P#	PNAME
P2	SCREW

PROPERTIES:

- **TABLES (values are atomic)**

  time varying tables,
  are not part of underlying mathematical theory.

- **KEYS (can be compound)**

  used for identification purposes,
  are not part of underlying mathematical theory.

INTEGRITY:

- **DEPENDING ON DBMS and VERSION.**

REFERENTIAL INTEGRITY

supplier

S#	SNAME	LOCATION
S1	JOHN	LONDON
S2	LEON	PARIS

supply

S#	P#	QTY
S1	P2	120
S1	P3	20

part

P#	PNAME
P2	SCREW

PRIMARY KEY:
the primary key of a table is a field or field combination of that table that can be used as an unique identifier for the records of that table.

Example: **S# in supplier table and (S#,P#) in supply table.**

FOREIGN KEY:
A foreign key is a field or field combination in one table whose values are required to match those of the primary key of some other table (or possibly of the same table).

Example: **S# in supply table (refers to S# in supplier table).**

ENFORCEMENT OF REFERENTIAL INTEGRITY
(C.J. Date 1986)

A RECIPE FOR PRIMARY KEYS (PK):

PK1. Specify NOT NULL for each field in the PK.
PK2. Create a unique index for each PK.
PK3. Ensure that this index remains in existence.
PK4. Keep PK specification in pseudo DDL.

A RECIPE FOR FOREIGN KEYS (FK):

FK1. Specify NOT NULL if NULLS NOT ALLOWED applies to fields in the FK.
FK2. Consider the merits of an index for each FK.
FK3. Use the authorization mechanism to prohibit all on-line SQL (i.e. query language statements)
- DELETES on the referenced table.
- UPDATES on the referenced table PK.
- INSERTS on the referencing table.
- UPDATES on the referencing table FK.

FK4. Keep FK specification in database maintenance programs.
FK5. Keep FK specification in pseudo DDL.
FK6. Use an utility program to check for constraint violations.