Chapter #3

Analysis & Reconciliation of Data Sources

Local source schema:

The schema of an individual data source.

⇒ Local source schema refers to the conceptual schema that describes the data stored in a source without regard to the models and technological solutions implemented.

⇒ focus on the representation of the application domain.

- Local source schemata result from the analysis of relations and constraints in relational database.

Global schema: A unified schema that represents the data from all sources.

Definition:

- Refers to the processes used to evaluate, compare and ensure consistency, accuracy, and integration of data from multiple sources before consolidating them in a unified system, such as data warehouse.

(i) Data source analysis:

Data sources can show strong relationships or can be completely indepen

Def: The process of identifying and evaluating the available data sources.

The Data most designer's primary goal is to improve his or her knowledge of data sources in the source analysis phase.

- key question involves:
  - What are the data types, formats and relationships in the data sources?
  - Are there inconsistencies or missing data that need to be addressed?
  - Are there duplicates records or conflict information across sources?
  - How does data from one source relate to data from another sources?

(ii) Reconciliation process:

⇒ Reconciliation process is needed in order to obtain consistent, error free information.

⇒ Reconciliation process involves:

- integrating
- cleansing
- transforming

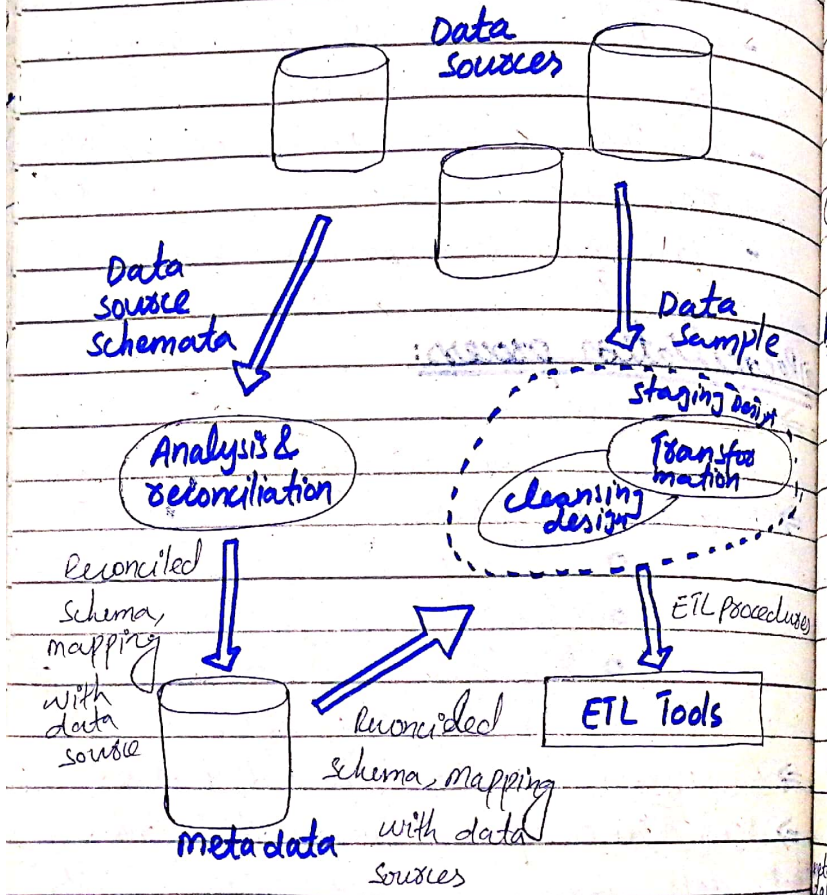
data to create a reconciled database.

⇒ This process requires time and resources.

⇒ Reconciliation process ensures that ensuring that data from different sources matches and correctly integrated.

⇒ Integration phase is based on the intensional features of data sources.

Three-Phase Reconciled layer design from data source layer:

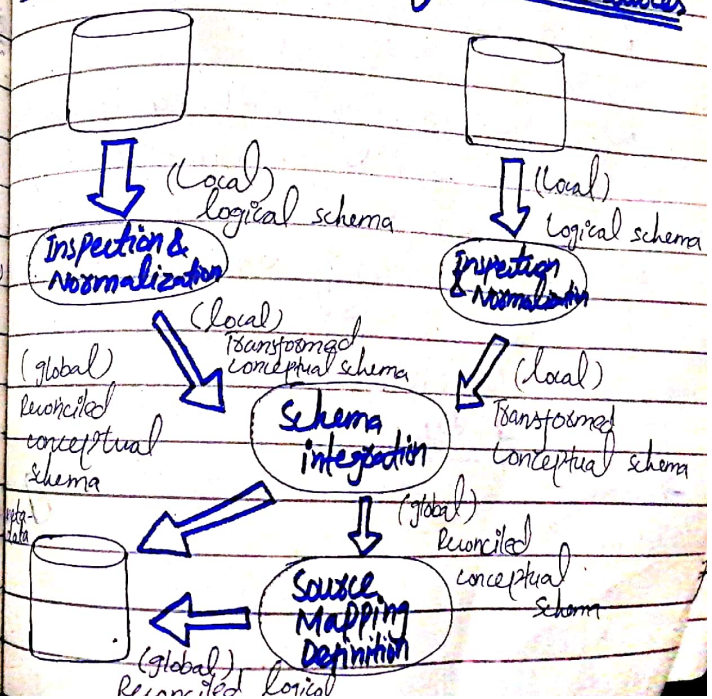


In short:

⇒ Reconciliation and analysis phase do the following:

- Inspect and normalize all the local schemata to generate a set of comprehensive, locally consistent, conceptual schemata.
- Then do integration, resulting in a globally consistent conceptual schema.

Design steps forming the analysis & reconciliation phase of two data sources



Inspecting & Normalizing Schemata

⇒ Detailed knowledge of data sources is necessary before data mart. conceptual design phase.

⇒ For doing so, need to perform the following tasks:

- Inspection
- Normalization

Inspection:

- Close investigation of local schemata.
- Refers to examining and understanding the structure, format, and quality of raw data before performing transformations.
- Goal is to identify any issues or patterns that might impact data integration into the data warehouse.

Key steps:

→ Data profiling – • Null values or missing data
- Inconsistent data formats
- Outliers or anomalies
- Duplicate Records
- Inconsistent naming conventions
- Data type verification of fields.
- Data integrity check
- Data consistency
- Error detection

Normalization:

- An attempt to rectify local schemata to model the datawarehouse domain as accurately as possible.
- Process of organizing data into structured format, before put into the data warehouses, where vast amount of data from different sources must be harmonized.
- Denormalization might be necessary in data warehousing to optimize query performance to speed up read operations.

⇒ In analysis phase, designers collaborate with the experts, such as data processing centers staff, managers etc.

⇒ Also they need to check or find any relationships that was accidentally missed or left.

for example,

Explicitly representation of functional dependencies skipped and new relationships between entities.

⇒ Both processes can brought changes in local schema for adjusting the specific data warehousing project needs.

⇒ All changes applied to local schemata, clarify explicitly all the concepts extracted from the data stored in data warehouse.

⇒ Normalization and inspection should be even single data source available

⇒ For more than one data sources, both processes repeated for every local schema.

⇒ Data designers should also check for which portion of local schemata nd need.

Example

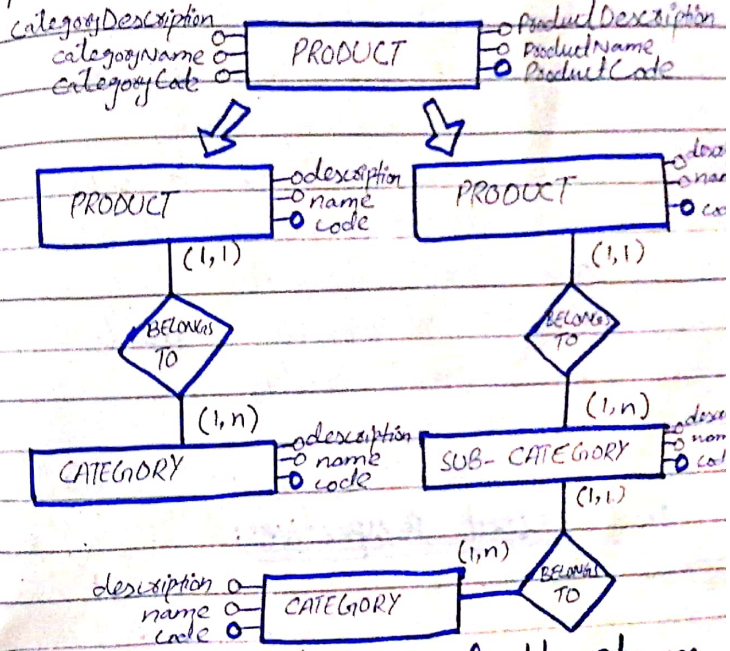
⇒ Suppose or Consider the ER-schema modeling of specific company production.

→ First change :

- ProductCode → CategoryCode can be apply bcz functional dependency is true.

→ Second Change :

- BELONGS TO association cardinality cannot be applied because the operational data does not contain the information revised to represent explicitly any products into subcategories.



- Show applicable & inapplicable changes in the source inspection & normalization phase

The Integration problems

→ Integrating a set of homogeneous data sources, such as relational databases, data files, and legacy sources, means detecting any relationships between the concepts represented in local schemata and solving the conflicts found to create one single global schema, whose elements can be mapped onto local schema elements.

→ if each data source modeled independently, then there will be no problem while integration process. However, this is not practically doing mostly.

→ The problems that need to be resolve during the integration phase. are:

- Different Perspectives
- Equivalent modeling Constructs
- Incompatible Specifications
- Common Concepts
- Interrelated Concepts

1. Different perspectives: (Different users may have different views of data)

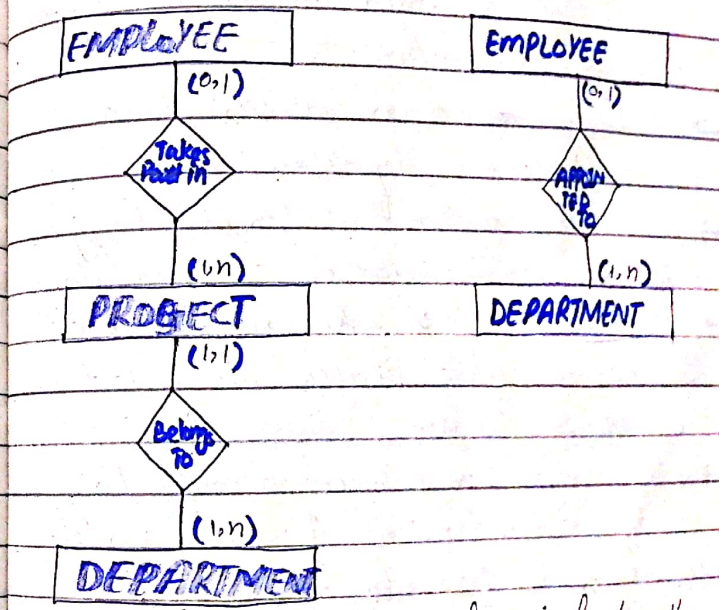
→ Users groups may have different expectations for how data should be presented and analyzed.

→ for example, Executives might need high-level summaries, while analysts might need detailed data.

→ Depending on the points relevant for tasks users have to perform need to be considered.

Example:

→ Considers to model the appointment of an employee to a department there Two perspectives:



→ The left hand modeling includes the project in which the employee is involved

→ This modeling could be suitable for

a database used for business organization chart management.

→ The right hand modeling does not include this concept because this is not relevant.

~~2. Employee~~

↑ (Different modeling techniques can represent the same information in different ways)

2. Equivalent Modeling Constructs

→ To represent individual concept, every modeling formalism can use different construct combinations.

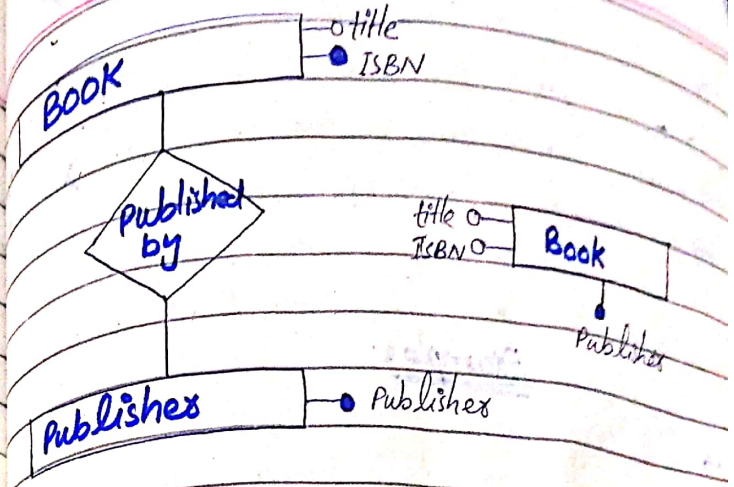
→ They may ~~use~~ not use identical data structures, terms or definitions.

→ This include variations in terminology, data organization, definitions and terminology.

→ for example, one system might use "Customer-ID" while another use "Client-No" for the same data.

→ To solve these issues, need to map and standardized the data models and transform data into a compatible format for cohesive integration.

Example:



→ The right hand side schema uses just one attribute, but the left-hand side aim at emphasizing the same concept creating the published entity.

→ Both modeling representations are perfectly equivalent, but merely syntactically different.

3. Incompatible Specifications

→ Issues arises when integrating data from various sources.

→ Incompatible specifications means different schemata modeling different, contrasting concepts.

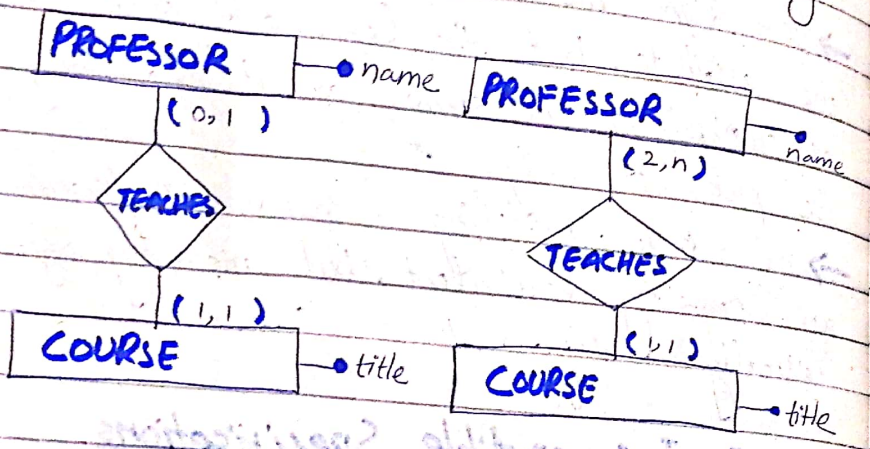
→ Differences can be due to:

→ Incorrect design decisions (Different naming conventions, Data types, Integrity constraints discrepancies →)

→ if data is time stamped in different time zones, they may reflect different versions and aligning this can be difficult.

Example:

→ Two unlikely modeling representations for the association between university professor and courses.



→ Both modeling representation could be result of error because:

- Unlikely that professor is not allowed to teach more than one course at the same time (on left side)
- He is required to teach more than on course (on right side).

4. Interrelated Concepts:

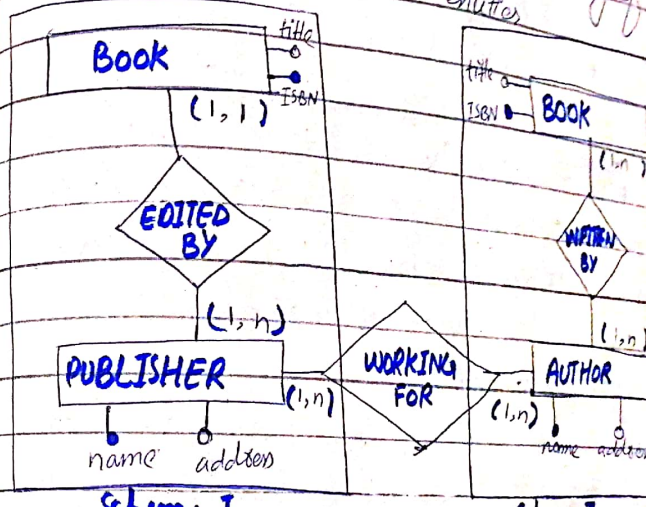
Once integration phase is completed, various interrelated concepts will be part of the same schema.

New relationships are created without types existing previous relationships.

These relationships are known as inter-schema properties and identified and represented explicitly.

Example

- An interrelation/interschema property of PUBLISHER and AUTHOR entities



Schema I Schema II

- Association between Author and publisher could not be represented in same schema.

5. Common Concepts:

→ Semantic relationships exist between common concepts that are modeled in different ways in separate schemata.

→ Four types of relationships exist b/w separate representation for the same concepts.

Identity:

- Two or more data records from different sources represent the same entity.
- R_1 and R_2 are exactly coincide.

Equivalence: (Two schemas are equivalent if they represent the same info in different ways)

- Occurs when two or more data elements from different sources contain identical information or values, even if represented in slightly different formats.

for example:

A product price stored as $100.00 in one system & 100 in another system is equivalent.

⇒ Two schemata R_1 and R_2 are equivalent if their instances can have one-to-one association.

if both schemas can be integrated without significant modification.

Compatibility:-

When R_1 and R_2 are neither identical nor equivalent but the constructs used and designer's point of views are comparable.

incompatibility:

- When R_1 and R_2 are incomparable because of inconsistent specifications.

eg: structural mismatches, semantic mismatches or conflicting values.

Two instances for the equivalent schemata:

Book

ISBN	title	Publisher

Publisher

Publisher

Book

ISBN	title	publisher

Conflict: → A conflict b/w two representations R_1 and R_2 portrays same concept every time both representations are not similar.

INTEGRATION Phases

→ After integrating local schemata, new features emerges and their representation requires complex set of tasks.

→ Different methodological approaches are used for handling those tasks:

1. Pre-integration
2. Schema Comparison
3. Schema Alignment
4. Merging & restructuring schemata

Preintegration

(This phase involves analyzing the data and identifying parts of the schema that can be integrated & discarded)

→ During this Phase:

- Data sources thoroughly analyzed
- Define the general integration standards.

→ Decisions taken during this stage are:

- what parts of schemata to be integrated? — bcz all operational data is not useful. Therefore, some parts of the schemata are scrapped.
- which integration strategy should be used?
- The integration techniques can be:

1. Binary Techniques
2. N-ary Techniques

N-ary technique:

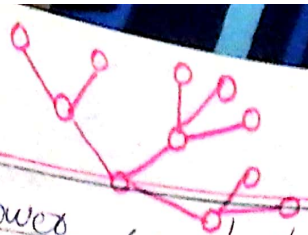
- more than two schemata integrated at the same time.
- Pros: Reduce the total amount of comparisons b/w concepts because each concept is only processed only once.
- N-Ary technique can be
  1. Single-step
  2. Iterative

1. Single-step:

- All N-schemata integrated at once.
- complexity is very high
- Used when have a clear understanding of all schemas and their relationships.

2. Iterative:

- All N-schemata are integrated incrementally not at once.
- And also mapped schemata gradually



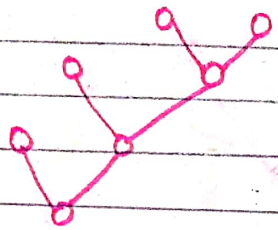
- Lower complexity
- Testing is done after each iteration.

Binary Techniques:

- Always involves pairs of schemata.
- Simpler than N-ary. Proper technique.
- Binary techniques can be:
  1. Ladder
  2. Balanced

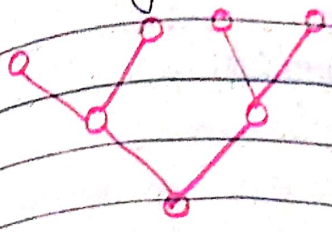
1. Ladder: (Prioritizing data sources and integrating them one-by-one).

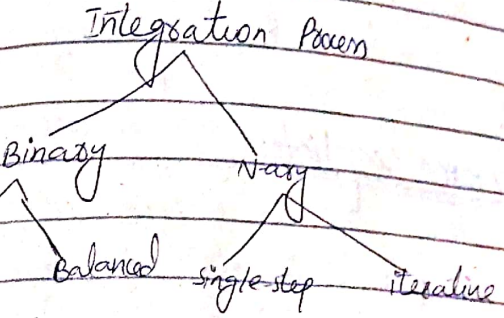
- New schemata are integrated into a current schema. Prioritize source schema & order in which they are analyzed
- Sequential and step-by-step strategy
- New or current schemata add the functionality of previous schemata and testing at each level before moving to the next.



2. Balanced:

multiple schemata integrated at the same time in parallel manner, also ensuring their pairs





Schema Comparision

(compares the schemas of different data sources to identify similarities, differences & conflicts).

→ This Phase consist of a comparision analysis of schema.
→ Relationships and conflicts discovered in schemata.
→ Effectiveness of this phase depends on the knowledge of designer that should do comparision btw different schemata. For this purpose they should collaborate.
→ Conflicts that are detected are:
  1. Heterogeneity conflicts
  2. Name conflicts

3. Semantic conflicts
4. Structural conflicts

1. Heterogeneity conflicts: → Due to differences in data structures & formats.

↳ Arise due to inconsistencies due to the differences in the structure of source schemata.

↳ Heterogeneity conflicts can be reduced by using common method for modeling the schemata.

2. Name conflicts: → Same data element is represented by different names in different schemas

↳ Arise due to different terms used by different designers in various data sources.

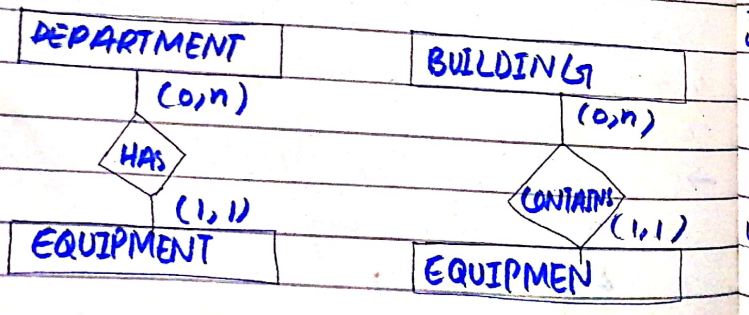
↳ Two main name conflicts occurs are:

1. Homonymies
2. Synonymies

↳ Same term used for two different concepts

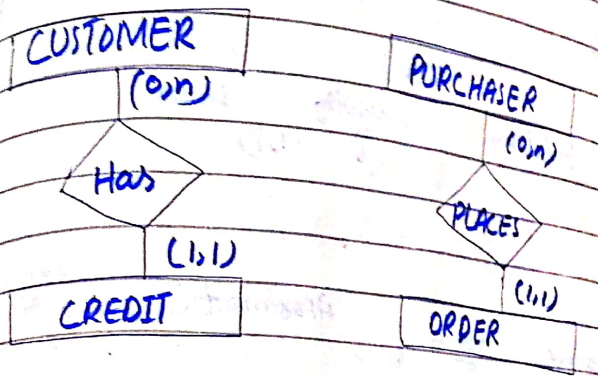
↳ Different terms used for same concept

e.g



problem occurs bcz EQUIPMENT left hand schema represent scientific equipment and right hand side for building furniture.

Reconciled schema should suitable make difference btw them.



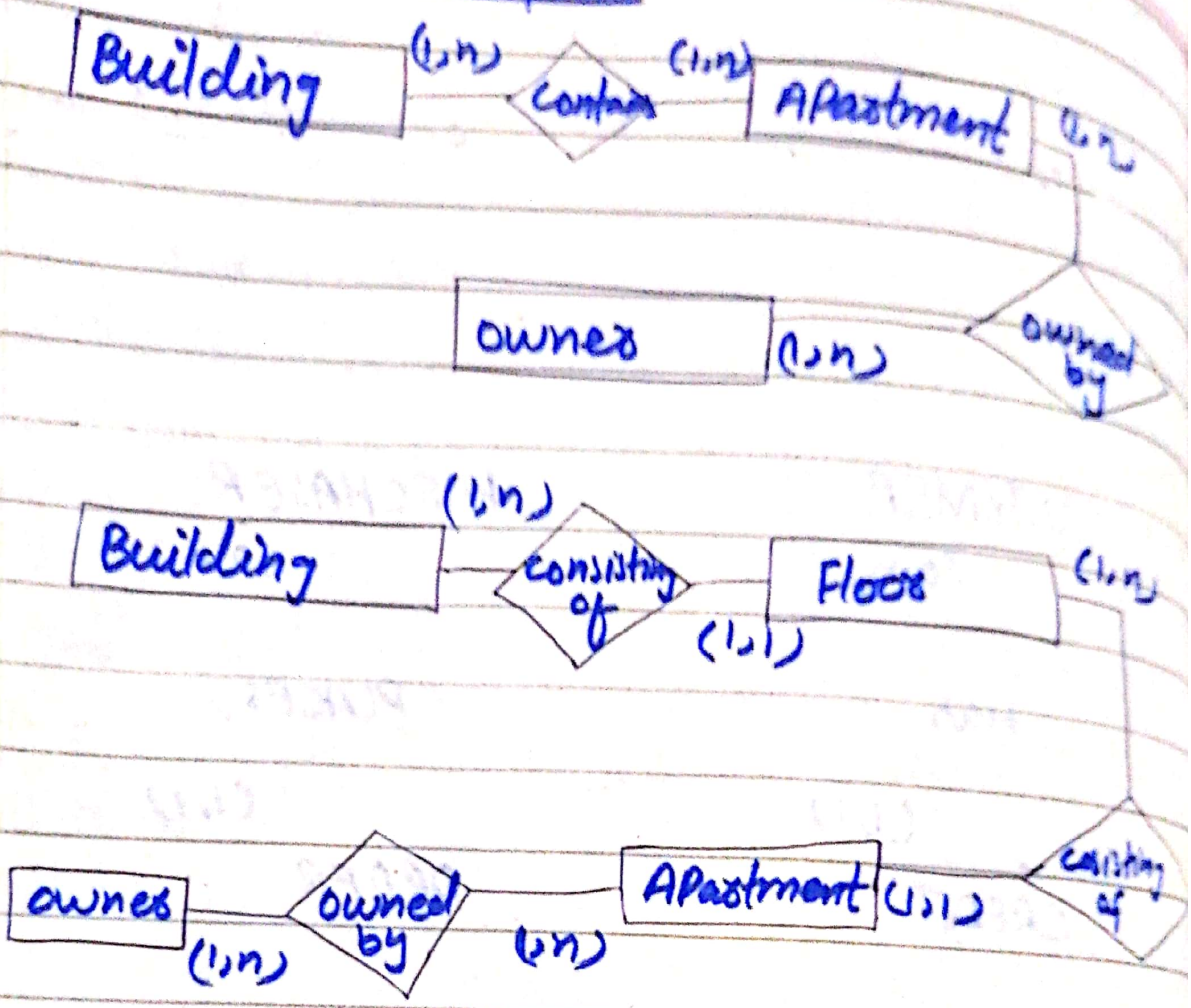
↳ Customers and purchasers cause synonym problem. bcz they refer to the same real world entity. Reconciled schema should avoid such type of duplication.

3. Semantic Conflicts:

↳ occurs when two source schemata represent similar concepts or entities in different ways & leading to misunderstanding when integrating. Arise due to

↳ similar data have different interpretations, meanings across different sources.

examples:



→ Two schemata in semantic conflict. they model a common part of the application domain at different level of abstract

4. Structural Conflicts:

→ Arise when different options for modeling the same concept or

→ Different integrity constraints are used

Categories into four types:

1. Type
2. Dependency
3. Key
4. Behavior

(on the next page)

ex: one schema may use VARCHAR for field while others use INT.

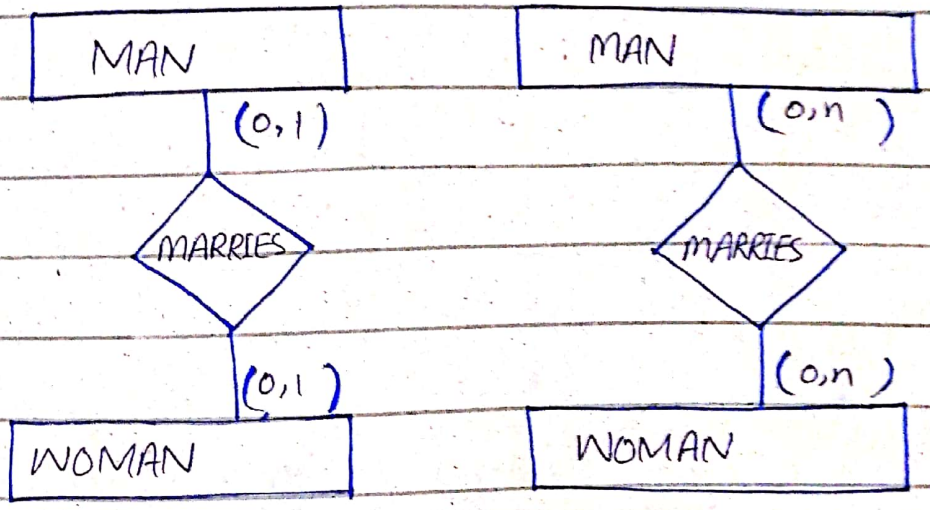
Type Conflict:

- Occurs when two different constructs are used to model the same concept.

Dependency Conflict:

→ there are contradiction in the rels b/w data elements in different schemas.

- Occurs when two or more constructs are interrelated with different dependencies in separate schemata.



(a) (b)

(a) define wedding between man and woman by one-to-one association.

(b) define wedding between man and woman by one-to-many association.

Key Conflict:

- Occurs when different attributes in separate schemata are used to identify the same concept.

for example,

⇒ People could be identified on the basis of their social security numbers or progressive numbering.

Behavior conflict:

- Occurs when different actions take to delete/edit data on the same concept in separate schemata:

for example:

Information on a customer is deleted from schemata if the customer has no pending orders, but another schema still stores the customer's information for future use.

3. Schema Alignment

(identifying & resolving conflicts of different schemas - ensures schema are compatible & can be merged effectively)

⇒ The purpose of this phase:

- Solve any conflicts occurs during previous step.

⇒ Can be do by applying transformation primitives to source schemata.

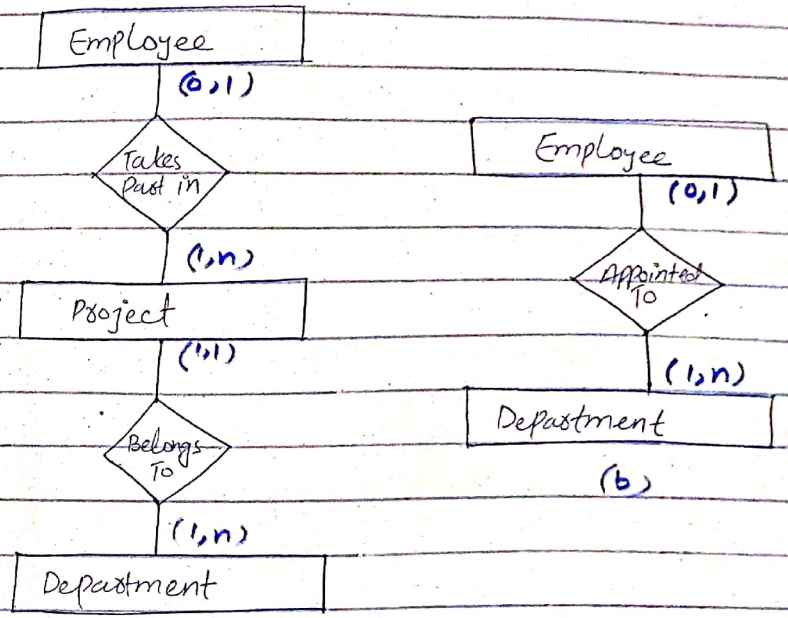
Transformation primitives:

⇒ Typically deal with:

- Changes in names
- Attribute Types
- Functional dependencies
- Existing constraints applied to schemata.

→ Conflicts cannot always be solved, because they stem from the basic inconsistencies of of the source database.

Example:



→ if semantic conflict occurs then, project concept is introduced on right hand side ~~of~~ schema.

Combining the compatible parts of the schema into unified schema.

4. Merging & Restructuring Schemata.

(requires to eliminate redundancy & improve data quality)

→ During this phase:

- All the aligned schemata merges into a single reconciled schema.
- After that, transformation process apply to improve the reconciled schema structure.
- For doing this, following properties need to check:

1. Completeness:

→ The merged schema should include all relevant attributes and entities from the source systems.

→ After overlapping source schemata, additional interschema properties may emerge.

→ A complete schema ensures that no data is lost.

→ Designers should determine new property that can be added by new associations or generalization hierarchies.

2. Minimality: (there are no redundant element).

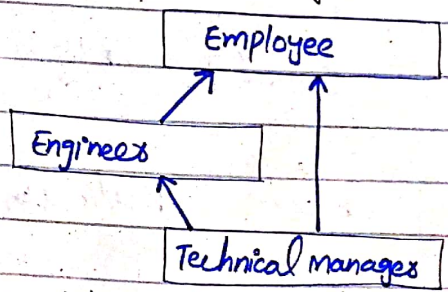
→ The merged schema should avoid redundancy and duplication of data, when overlap many schemata.

→ May be mutually derived in reconciled schemata.

for example:

→ Generalization between TECHNICAL MANAGER and EMPLOYEE is redundant and can be deleted.

→ Cyclic relationship between them also cause redundancy.



3. Readability: (refers to the clarity & understandability of the schema)

- The merged schema should be easy to understand and interpret.
- Use clear and meaningful attribute and entity names.
- Typically, it is difficult to quantify the readability differences b/w two schemata.

Defining Mappings

→ The result of source schema analysis targeted to integration consists of two elements:

1. Reconciled schema where the conflicts between local schemata are solved
2. The set of relationships between the source schema and reconciled schema elements.

→ This mapping process ensures that data is accurately transformed and loaded into data warehouse.

Two approaches that define relationship blw source system and the target data warehouse:

GAV

- Global as view defines the target data warehouse as a view over the source systems.
- This means that the target schema is defined in terms of SQL queries that extract from the source systems.

Pros: 1. Simplicity 2. Flexibility

↑ where the global schema is defined as a set of views of the local schema.

Local as View (LAV)

→ LAV defines the target datawarehouse as a collection of materialized views that are populated from the source systems.

→ These materialized views are physical tables that store the transformed and aggregated data.

→ This approach requires very complex transformation processes, generally called rewriting.

Example of Mapping

Source: A customer table with fields: customer_id, first_name, last_name, address.

Target: A customer dimension table with fields:
customer_key, customer_name, address_line1, address_line2.

Mapping Rules:

1. Customer_id from source maps to customer_key
2. Concatenate first_name & last_name from source to create customer_name in target.
3. Split address from source into address_line