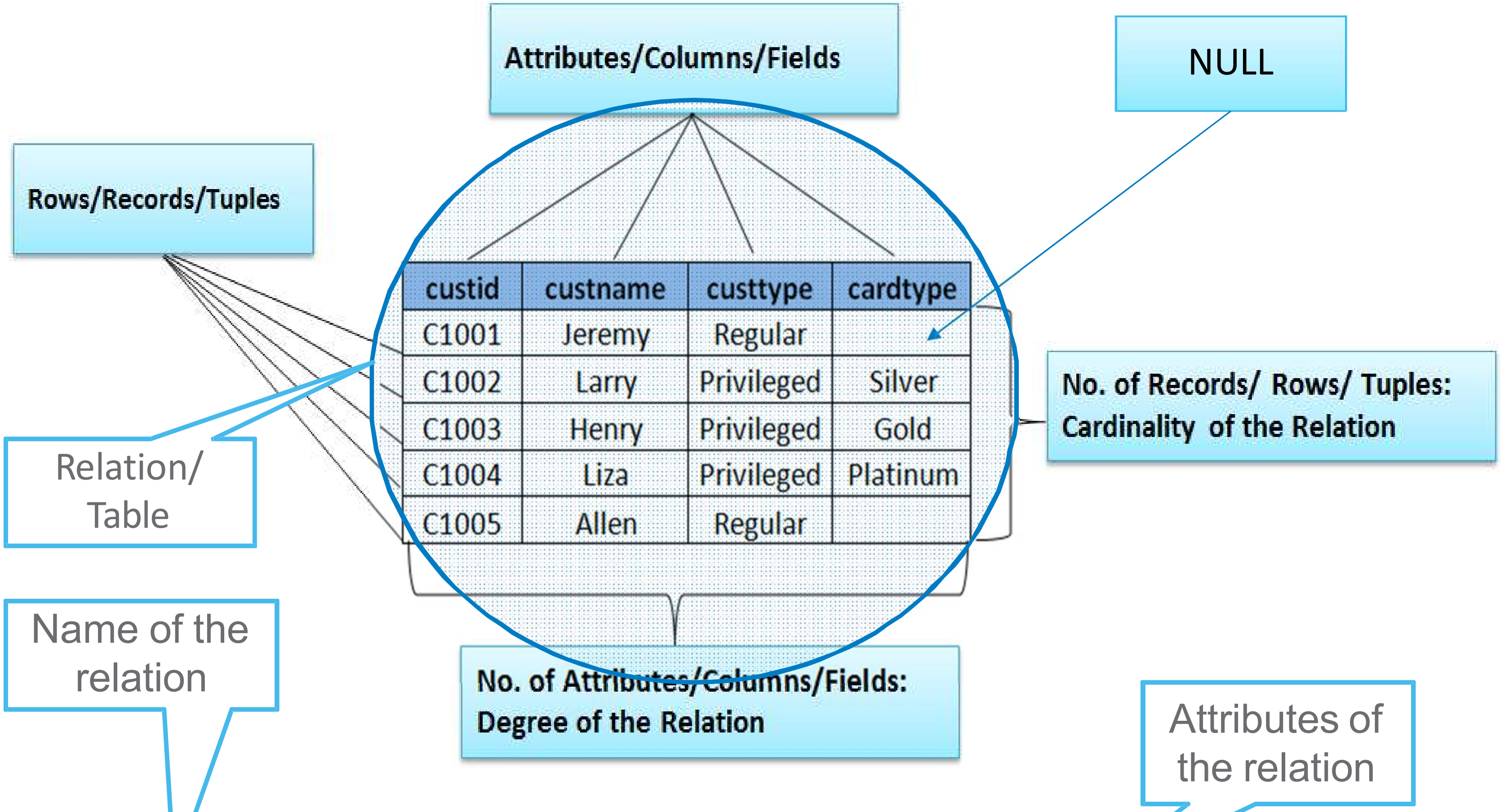
Conversion of entity relationship diagram to relational schema

Data representation in RDBMS(Revisit)

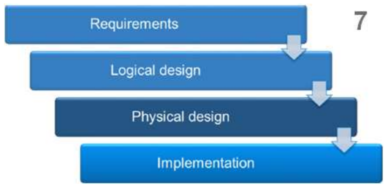

Attributes/Columns/Fields
NULL
Rows/Records/Tuples
Relation/ Table
No. of Records/ Rows/ Tuples: Cardinality of the Relation
Name of the relation
No. of Attributes/Columns/Fields: Degree of the Relation
Attributes of the relation

custid	custname	custtype	cardtype
C1001	Jeremy	Regular	
C1002	Larry	Privileged	Silver
C1003	Henry	Privileged	Gold
C1004	Liza	Privileged	Platinum
C1005	Allen	Regular	

Relation is usually represented as:
customer (custid, custname, custtype, cardtype)

Confidential

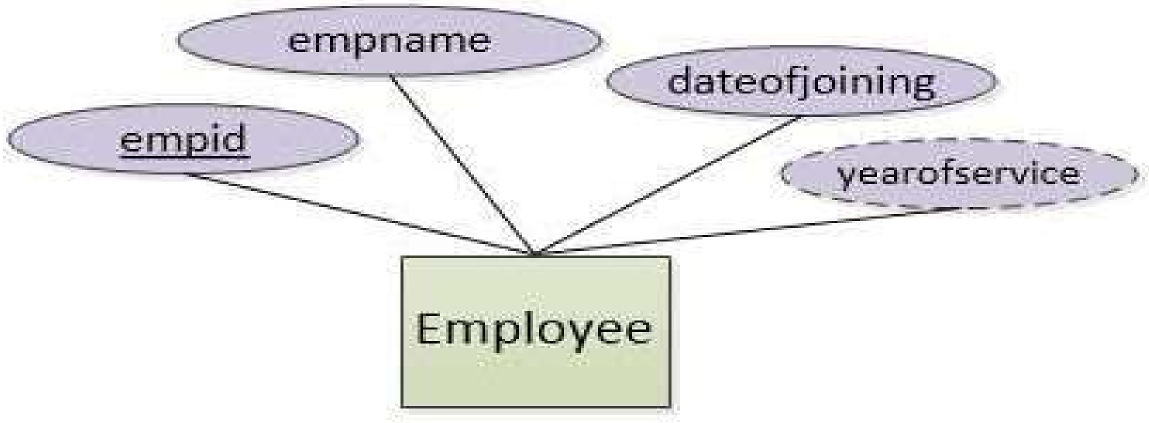
Conversion of ER model to relational schema



- **Schema**
  - A description of a database
  - Specifies the relations, their attributes and the domains of the attributes
- **Conversion guidelines**
  - Each entity in ER diagram becomes a table in relational schema
  - Each single-valued attribute in ER diagram becomes a column of the table
  - Derived attributes of entities are ignored
  - Composite attributes of an entity are represented by its equivalent parts
  - Multi-valued attributes are kept in a separate table
  - The key attribute of an entity is chosen as the primary key of the table
  - Converting relationships is based on degree and cardinality of relationship

Confidential

Example: Strong entity (1 of 2)

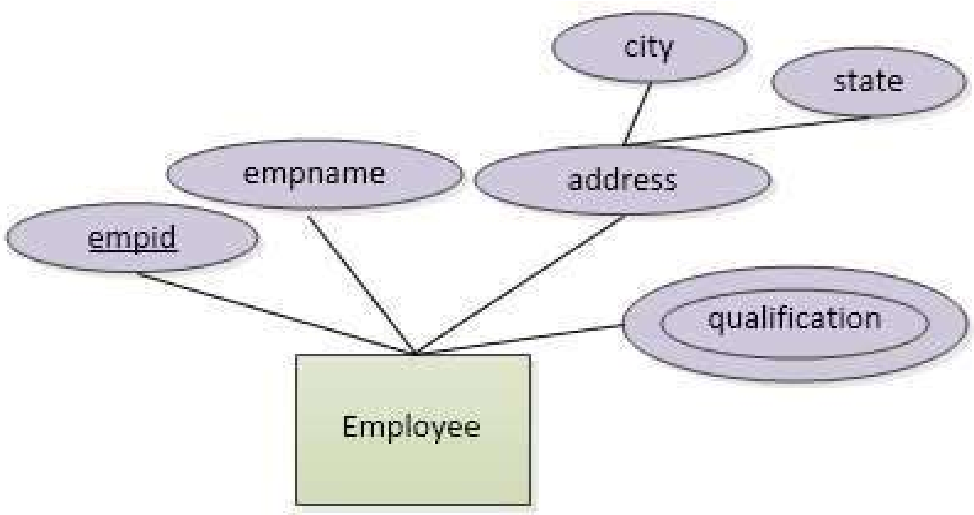


Relational Schema:

employee (<u>empid</u>, empname, dateofjoining)

Confidential

Example: Strong entity (2 of 2)



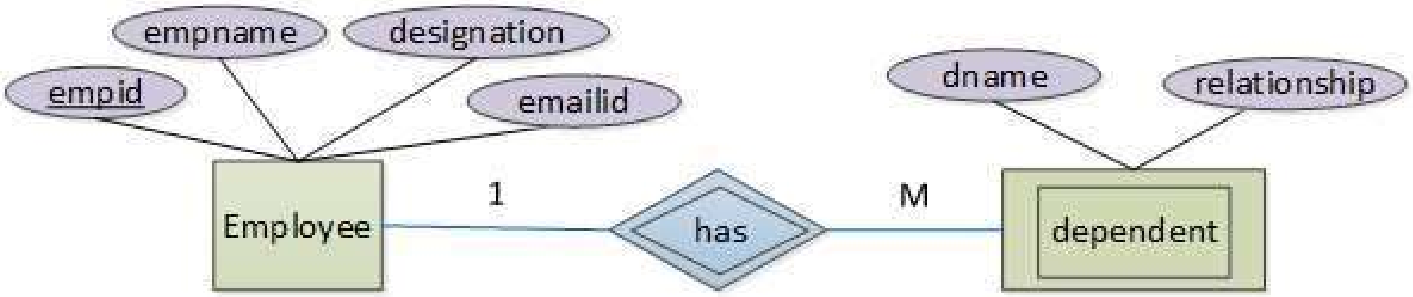
Relational Schema:

employee(empid, empname, state, city)

employeequalification (empid, qualification)

Confidential

Example: Weak entity



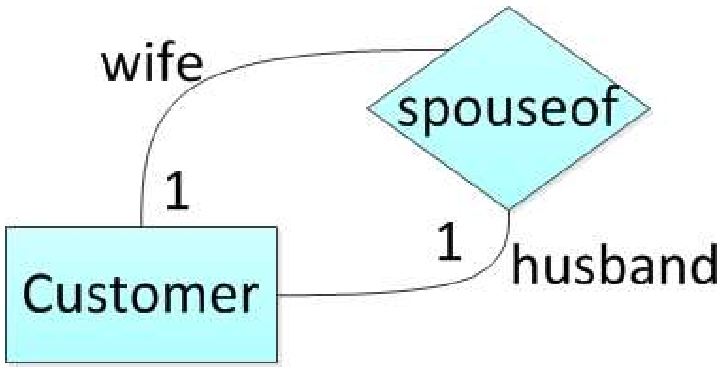
Relational Schema:

employee(empid,empname, designation, emailid)

dependent (empid, dname, relationship)

Confidential

Example: Unary 1:1



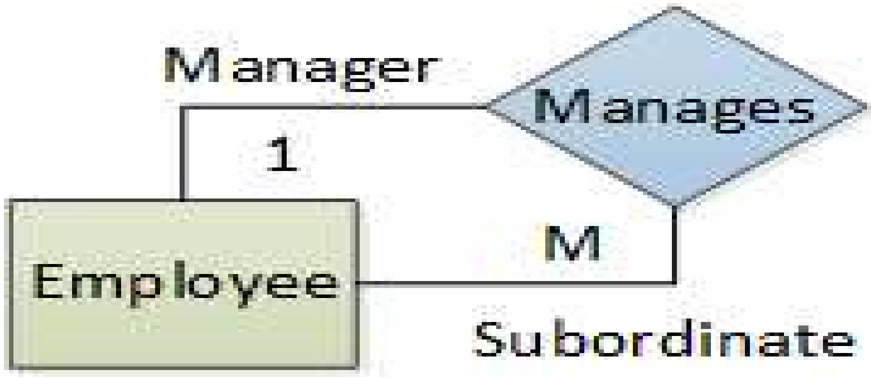
Relational Schema:

customer(customerid, customername,…spouse)

Confidential

Example: Unary 1:M

- The primary key of the table will itself become foreign key of the same table

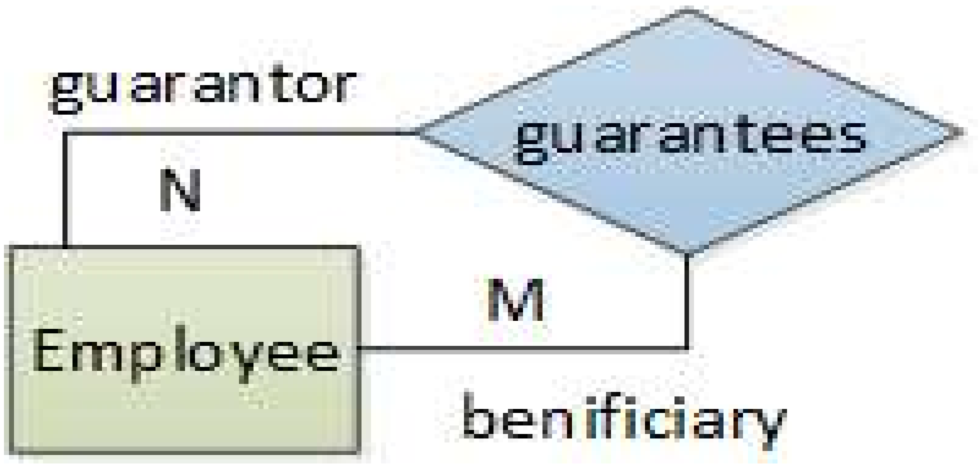


Relational Schema:

employee(empid, empname, designation, ..., managerid)

Confidential

Example: Unary M:N



Relational Schema:

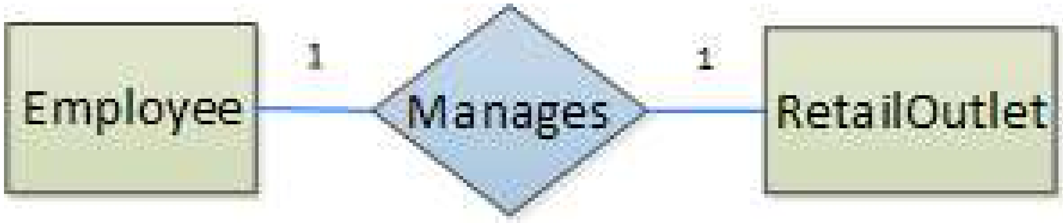
employee(empid, empname, designation,….)

guaranty (guarantorid, beneficiaryid)

Confidential

Example: Binary 1:1

- The key attribute of any of the participating entities in a relationship can become a foreign key in the other participating entity



Relational Schema:

employee(empid, empname, designation, ……………., salary)

retailoutlet (retailoutletid, retailoutletlocation, retailouletmanagerid)

OR

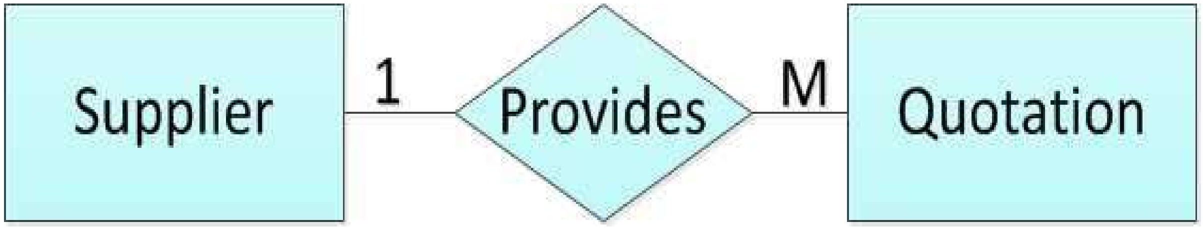
employee(empid, empname, designation,…….,salary, retailoutletid)

retailoutlet (retailoutletid, retailoutletlocation)

Confidential

Example: Binary 1:M

- The key attribute of entity on the “1” side of the relationship becomes a foreign key of entity towards the “M” side



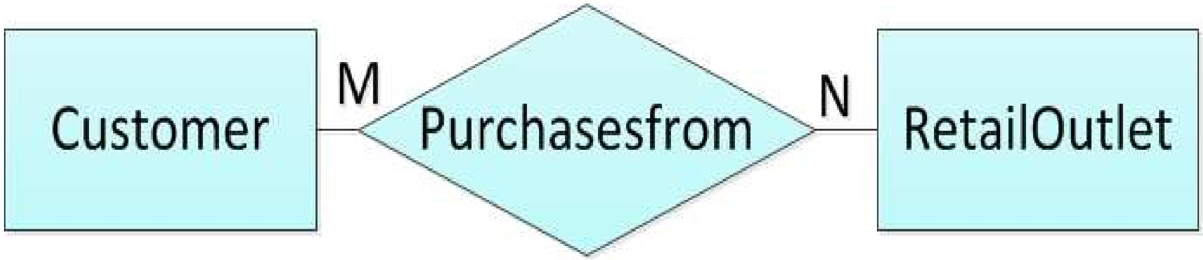
Relational Schema:

supplier (supplierid, suppliername, suppliercontactno, supplieremailid)

quotation (quotationid, itemcode, quotedprice, supplierid)

Confidential

Example: Binary M:N



Relational Schema:

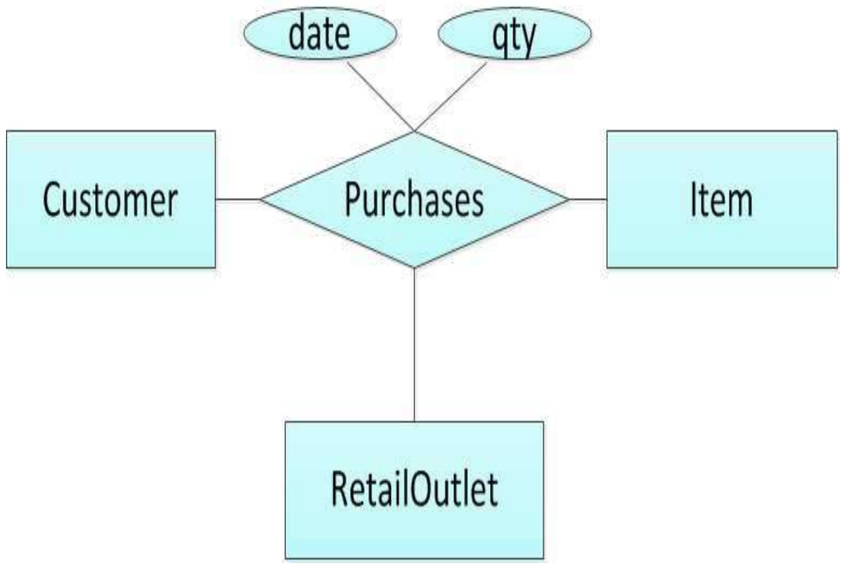
customer (customerid,customertype,customername,emailid,contactno,address)

retailoutlet (retailoutletid, retailoutletlocation)

purchasesfrom (customerid, retailoutletid)

Confidential

Example: Ternary relationship



Relational Schema:

customer (customerid,customertype,customername,emailid,contactno,address)

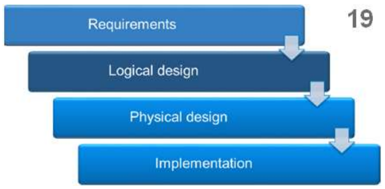
retailoutlet (retailoutletid, retailoutletlocation)

item (itemid, description, reorderlevel, itemclass)

purchases(customerid, retailoutletid, itemcode, date, qty)

Confidential

Database life cycle – Logical design



Top down approach (Entity – Relationship (ER) model)

- This approach is used when application **requirements are clear**
- Represents the application requirements in a pictorial form
- The real world objects and their corresponding attributes are identified from the requirements – hence it is top down
- This model helps in
  - analysis and design
  - re-validating the requirements

Bottom up approach (Normalization)

- This approach is used when application **requirements are not very clear**
- First define the required data items and then group the related data items
- Further refinement may be carried out depending on the application need

Confidential