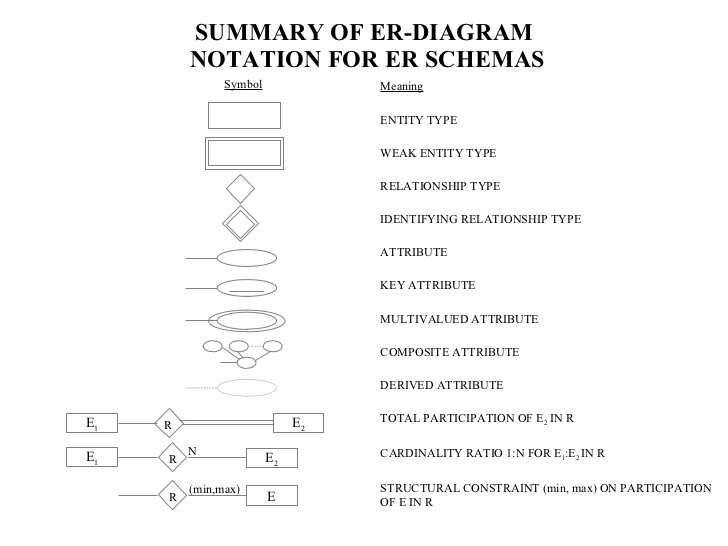
>>Schema- description of the database. It specifies the structure of various entity and relations in a DB.

>>In a DB  
 Entity=Table  
 Attribute=Column  
 Tuple= Row  
 Relation=Joining of tables

Types of Attributes  
  
>>Multivalued – Column (address) will have multiple values like {Bangalore, Chennai, Mumbai}  
>>Composite- Column (address) will be made of various atomic values like {City, H.NO, PIN}  
>>Key attribute- value is unique to every row  
>>Derived attribute- {DOB}-->age

ER-Entity Relationship Diagram  


>>Degree of relationship- how many entities(tables) are joining to form that relation. {Binary, ternary}

>>Cardinality ratio- {1:1, 1:N, N:M}- it is only for binary relations

>>Participation in a relation- {Full, partial}

Functional Dependency  
  
In a entity/table - [X,Y = set of attributes, t1,t2 are tuples]  
if X->Y then  
{  
if(t1[X]==t2[X]) //tuple 1 has same values for X(maybe set of 3 columns) as tuple 2  
then  
t1[Y]=t2[Y] // then tuple 1 will have same values for Y(maybe set of 2 columns) as tuple 2  
}

KEYS

>>There are various candidate keys that can uniquely define a tuple, one of them becomes primary key.  
>>Superset of any candidate key is called super key.  
>>Attribute part of candidate key is a prime attribute.

RULES

IF THEN

X is superset of Y X->Y  
X->Y XZ->YZ  
X->YZ X->Y AND X->Z  
X->Y AND Y->Z X->Z  
X->Y AND WY->Z WX->Z

NORMALIZATION  
  
Schema is checked through various tests if it passes those test it is in NORMAL FORM, else changes need to be made.

1NF- attribute should be atomic  
Soln-

>>1NF with redundancy  
ID Address  
123 a1  
123 a2  
123 a3  
456 xyz  
789 ghj  
  
>>1NF with breaking the attribute  
ID ADDRESS1 ADDRESS2 ADDRESS3  
123 a1 a2 a3  
456 xyz NULL NULL  
789 ghj NULL NULL

2NF- fully function dependency  
  
X with columns {x1,x2,x3}  
X->Y is fully functional dependent iff {x1,x2,x3} collectively define Y !  
  
if x1->Y or x1,x2->Y then not fully functional dependent  
  
Soln- break the table into more tables such that it becomes in 2NF form i.e. fully functional dependency.

3NF-  
X->A  
if X is a super key OR A is a prime attribute i.e. if anyone of these is true, it is in 3NF.  
  
BCNF-  
X->A  
if X is a super key

Example-  
{A,B} is primary key  
C is a non-prime attribute.  
{A,B}->C; C->B; satisfy 3NF but not BCNF  
  
3NF is a superset of BCNF

Things to take care about while designing DB in Relational DBMS  
>>No spurious tuples (caused by joining without proper condition)  
>>No Redundant information(same info stored in different tables, dbs). As when you have to update, delete, insert at one place you then must modify same thing in different places as well. Thus causing update, delete, insert anomalies.  
>>Should be in normal form.

Foreign key- primary key of one table defined in another table.  
  
Referential integrity- Referential integrity is a database concept that ensures that relationships between tables remain consistent.  
Primary key of t2 is referenced in t1 as a foreign key.  
  
T1 ={pk\_t1, col1, col2, col3, fk\_t2}  
T2= {pk\_t2(becomes fk\_t2), c1, c2, c3}  
  
When one table(t1) has a foreign key to another table(t2), the concept of referential integrity states that you may not add a record to the table(t1) that contains the foreign key unless there is a corresponding record in the linked table(t2).  
It also includes the techniques known as cascading update and cascading delete, which ensure that changes made to the linked table(t2) are reflected in the primary table(t1).

CASCADING- pass (something) on to a succession of others.  
So if something is updated or deleted in T2 then it is updated/deleted in T1 as well  
if CONSTRAINT: Cascading on update/delete is applied.  
These constraints are called FOREIGN KEY CONSTRAINT.

Apart from cascading there are other options too like – NULL on update/delete.

-------------------------------------------------------------------------------------------------------------------------------------