**Relatii**

* **Entity Relationship Model**(Model de asociere cu entitati),care descrie cu cuvinte:
* numele entitatilor
* numele atributele
* poate si primary key
* cum se asociaza relatiile(de ex Un sofer poate avea mai multe masini si o masina trebuie sa apartina unui sofer), se reprezinta prin **Entity relationship Diagram**, si e un concept **abstract**.

Nu exista un standart pentru ce trebuie sa contina Entity RelationShip Model

* Pentru a crea o **baza de date** fizica, **Entity Relationship Diagram** trebuie convertit in **Modele Relationar**e pentru a le implementa ca baze de date fizice.
* Orice relatie(tabel) are:
* **Nume**
* **Gradul relatiei** – numarul de coloane(attribute)
* **Cardinalitatea relatiei** – nr de obiecte,linii
* **Domeniu** – set de valori premise pe care le pot lua unul sau mai multe attribute
* **Tuple** – obiect concret(linie)
* **Relation Database** – colectie de tabele cu nume distincte.

**Keys**

Intr-o relatie putem sa avem diferite tipuri de identificatori. Pentru a-I descrie, **se** folosesc mai multe tipuri de chei.

* **Superkey** – atribut sau multime de attribute care pot **identifica in mod unic o tuple intr-o relatie sau care fiind folosite, pot afla valorile celorlalte attribute din tuple**.E ca un identificator simplu, doar ca nu are limita la numarul de attribute. Superkey poate reprezinta numarul **maxim** de attribute ce pot identifica in mod unic un tuple. Putem folosi unul ,2,3 si mai multe atribute.Macar si toate attributele din relatie. Nu e limita la folosirea lor, atat timp cat multimea de attribute identifica unic un tuple,adica macar un atribut din ele poate identifica obiecte unice, sau(mai rar) mai multe atribute, sau chiar toate odata o pot face. Fie ca avem o multime de attribute, si daca unul din el poate identifica o linie in mod unic din tabel,sau daca trebuie sa folosim mai multe din ele neaparat(fara a exista unul ce singur poate sa identifice,desi e rar), multimea e deja considerata super-key, nemai contand ca sunt si attribute ce singure nu pot identifica in mod unic un rand. Nu e limita la nr de attribute, atat cat ele impreuna pot identifica un tuple, chiar daca unele sunt in plus sau sunt si mai multe ce singure ar putea sa identifice un tuple. **Superkey e doar un concept theoretic**. O multime de attribute ce nu pot identifica un tuple nu e super key! **Multimea poate avea si un atribut!**

De exemplu, daca avem atributele ID, IDNP, Nume, Prenume,

Pentru a identifica in mod unic o linie, am putea folosi ID, IDNP. Numele si Prenumele nu garanteaza ca vor gasi un rand unic, nici impreuna. Deci orice multime in care participa ID sau/si IDNP e super key, de ex:

{ID} {ID, IDNP} {ID,NUME} {ID,NUME,PRENUME} {ID,IDP,NUME,PRENUME} si tot asa.

**{Nume,Prenume} – nu e super key!!!**

**{Nume, Prenume, Data Nastere} ar putea fi super key in alt tabel, dar nu e o idee prea buna.Se Evita asa ceva.**

In orice superkey exista cel putin o canditat key!

* **Canditate key** – reprezinta dimensiunea minima a unei superkey. Reprezinta multimea de attribute ce contin numarul minim de attribute ce pot identifica un tuple,de obicei fiind formata dintr-un singur key, fara attribute in plus .

De exemplu, daca avem atributele ID, IDNP, Nume, Prenume

Candidate key pot fi ID si IDNP doar, si atat!

Nu mai putem crea alte multimi, caci deja ar fi prea multe elemente, dar candidate key e minimul de attribute sau un atribut chiar ce identifica o linie

De ex, {ID, IDNP} nu pot fi candidate key, caci din multimea data mai putem forma submultimi de candidate key, sau {ID, Nume} nu e candidate key caci Nume e in plus.

Uneori putem avea nevoie de cel putin 2-3 atribute pentru a identifica o linie unica, si in asa caz multimea lor va forma o keie candidata.Ex {Nume, Prenume, Data Nastere} ar putea fi candidte key, dar nu e o idee buna.

Orice candidate key e un super key, dar nu orice superkey e un candidate key!

* **Primary key** – candidate key aleasa pentru a identifica un tuple in mod unic intr-o relatie. Poate implica si mai multe attribute. Fiecare relatie are doar o primary key!
* **Alternate Key** – candidate keys care nu sunt alese ca primary key
* **Foreign key** – Atribut sau set de attribute, dintr-o relatie, ce realizeaza o conexiune cu o alta relatie.

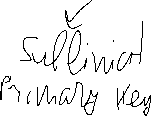
**Aceasta mereu este** o primary key din alt tabel ce se conecteaza cu acesta, in care ea e candidate key .

* Cel mai important lucru intr-o relatie e de a gasi primary key, care trebuie sa fie minim mereu si cea mai potrivita.

**Schema relationala**

Defineste o relatie prinrt-un set de attribute si domeniul lor.

Forma generala: NumeTabel(Atribut1, Atribut2,…,Atributx(FK),.., AtributN)



**Schema relationala a bazei de date**

Un set de scheme relationale, fiecare cu un nume propriu.

**Model Relationar**

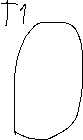
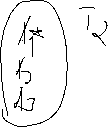
O multime de scheme relationale

**Proprietati ale relatiilor**

* Fiecare tuple e distinct, neexistand tuple identice!
* Ordinea atributelor,tuplurilor nu are importanta
* Fiecare celula contine exact o valoare
* Fiecare atribut are un nume unic intr-o relatie.
* Valorile unui atribut sunt dintr-un domeniu posibil.
* Relatia are nume unic in baza de date.

**Convertirea la Entity RelationShepDiagrams/Models in Modele Relationare**

1. **Entitatea** e transformata in tabel(relatie)
2. Atributele entitatii devin attribute ale relatiei
3. Identificatorul entitatii devine primary key in relatie.
4. Asocierile se marcheaza ca foreign key.

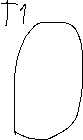
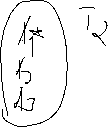


* Daca avem 1 la N ca relatie, tabelul al doilea trebuie sa aiba ca Foreign Key Primary key din tabelul 1.

T1(A1, A2, A3)



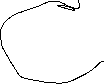
T2(A1, A2, A3, A4(FK))



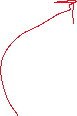
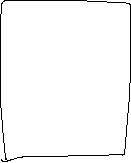
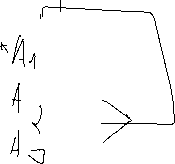
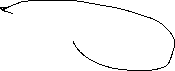
* Daca avem N la M, trebuie sa cream o relatie noua formata doar din Foreign Keys, care sunt Primary Key din ambele relatii ce **vor fi Primary key impreuna si in noua relatie**,si separat si foreign key, si cu numele format din cele 2 tabele

T1(A1, A2, A3)

T2(A1, A2, A3)



T1\_T2(A1\_1(FK), A1\_2(FK))



* Daca avem o relatie **unara**, sau I se mai spune recursive:
* 1 la N - se creaza un nou atribut ce va fi foreign key, dar ce va fi identic cu primary key, doar ca va avea alt nume(cel din asociere, deobicei)

T1(A1, A2, A3, A4(FK))



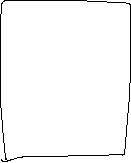
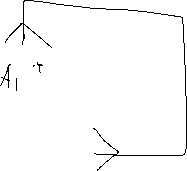
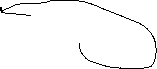
* N la M – cream un tabel nou cu 2 atribute, identice cu primary key din tabelul de anterior, ele fiind si aici primary key, si foreign key, doar ca al doilea difera ca nume, ca sa nu fie identice.

Numele tabelului e compus din numele primului tabel + \_ + iar numele primului tabel

T1(A1, A2, A3)



T1\_T1(A1(fk), A2(fk))



* **1 la 1**

**Le unim si face o singura relatie si gasim o primary key.**

**Nu le unim si:**

* Le tratam ca 1 to M
* Daca ambele tabele sunt obligatorii, il alegem pe unul ca 1 si altul M
* Daca unul e obligatoriu, il tratam ca 1 si celalalt M
* Daca ambele sunt optionale, trebuie sa regandim primary key

Daca Foreign Key e si primary key, se subliniaza(cand cream un nou tabel la relatia N la M, foreign keys sunt si primary):

Tabel\_Tabel(A1(fk),A2(fk))

Daca asocierea are mai multe entitati, trebuie cumva sa le separam ca sa avem unary/binary asociere, adica 1 sau 2 relatii conectate. Daca nu le separam cumva, nu putem face nimic. Bazele de date relationale nu prea pot sa se ocupe de mai mult de 2 relatii