13363332 Bao Hoang

Basics

- Components of Db

```
DataModeling/Design tools     Software(UI)
Repo                          DB Admin/Developer
Db,DBMS                       End users
```

  - MetaData: `datadype`, `file size`, `allowable values`, `data context`
- Ansi-Sparc Architecture
  - Conceptual(ERD)
  - Internal: : logical Datamodel(ERD), phys
- **Business Rules** are often used to determine **entities**, **attributes**, **attribute types**, **identifiers**, **relationship**
  - BR are Declarative, precise, atomic, consistent, expressable, distinct, business oriented
  - **Attr** `simple`, `composite`(address), `multivalued`(1:m)
  - **Keys** simple, compositem, candidate(combination of attr see **Relation**)

ERD

- **Degree of Relationship**: Unary, Binary, Ternary
- Cardinality `1:1`, `1:m`, `m:n`(associative entity)

```
e.g product, supplier, store
  **Store** must have one or many **products**, 1 supplier supply from
many produtcts
```

- `StrongEntity`(independant), `WeakEntity`(PK depends on Strong entity)
- Solid(weak-strong), Dotted(regular)

Super/Sub (Inheritance)

- Process of building sub/supertype entities
  - **Generalization**(Bottom-up, find comonalities), **Specification**(top-down)
- **Constraints**
  - Completedness
    - `Total`(DoubleLine) has to be one subtype
    - `Partial`(SinlgeLine) maybe a subtype, maybe not

    - Disjointness
        - `Disjoint` supertype can be only ONE of the subtype
        - `Overlap` supertype could be MORE than one of the subtype

Relation

- integrity constraints: DeleteRules
    - `Restrict`(dont delete if exist in parent), `Cascade`(auto delete any correspondent), `Set-to-Null`(Ser null reference, not allowed for weak/assiciated)
    -

Normalization

- To avoid **anomalies**: Insertion, Deletion, Update
- Functional Dependancy `X -> Y` X determinant, y Dependant
- **Keys**
    - `Super` set of attr can be use to id a row(instance)
    - `Candidate` minimal set of attr can id an instance, part of the candidate will become PK
- Determinind candidate keys with Functional Dependancies
    - LMR table using FDs
    - Determine Closure (see example Question)
- **Partial Transistive FDs**
    - `Partial` FD where non-keys are dependent on a part of the Composite
    - `Transistive` FD where attributes are determined by another non-key attr
- **Normalization Form**
    - `1NF` no derrived, no multivalued attributes, all attr are atomic
    - `2NF` **no partial dependancies**, every non-key must be defined by the whole key
    - `3NF` **no transistive dependancies**(fd on nonkey), split retalion to fix

Sample Question

Consider These `A→ D AE → H DF → B, C E → C H → E`

```
21. Which of the following dependencies is implied by above dependencies?
A) A → DH            B) AED →C           C) DA → C           D) ADF → E
C because AE → H and H → E and E → C therefore AED →C.
```

```
22. Consider the decomposition into 4 relations: R1(AEH), R2 (BCDF), R3
(AD) and R4(CE). This decomposition is in: A) 1NF     B) 3NF       C) 2NF
D) None of the above

**Check R1(AEH) related FDs:  AE → H & H → E** draw LMR

A+={A} which is not equal to R
AE+={AEH} =R1 so AE is a candidate key.
AH+={AHE} =R1 so AH is also a candidate key.
```

The minimum combination of the attributes that can determine all attributes in a relation are the candidate keys of that relation. AE+=R1 and AH+=R1 so AH and AE are candidate keys of R1.

Considering that AE and AH are both candidate keys.
If you consider AE as PK then R1(AEH) will be in 3NF.
If you consider AH as PK then R1(AEH) will be also in 3NF.

Check R2(BCDF) related FDs which is just DF → B, C Draw LMR
DF can be PK, check NF rules > 3NF

same for R3 R4. 3NF

23. Consider the decomposition into 3 relations: R1(AD), R2(HEC) and R3(BCDF). This decomposition is in:
A) 1NF B) 3NF C) 2NF D) None of the above

R1, R3 3NF R2(HEC) 2NF
H-> E E-> C

H is the Candidate/PK
Since E(nonkey) is a determinant for another C(non-key) > Transistive Dependancy > Violate 3NF
R2 is 2NF

25. What is the highest normal form of Relation R(A, B, C, D, E, F)?
A→F BC→E BC→D E→F BC→F

ABC is the Candidate
R = 1NF
R !=2NF because
A -> F
BC -> F
non key determined by part of key (**partial dependency**)