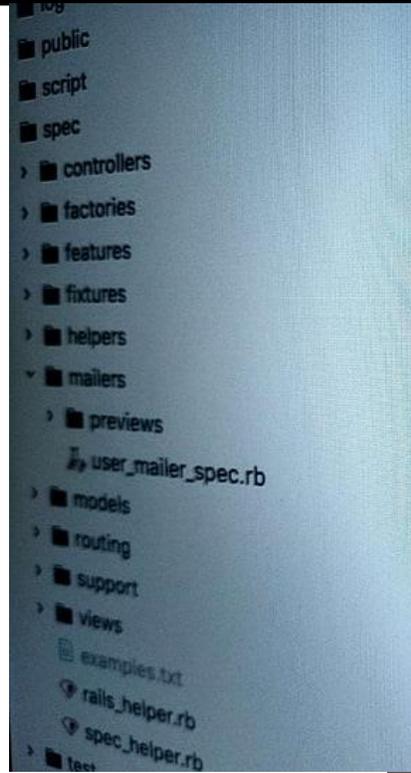


# தரவுத்தள முகாமைத்துவ முறைமை

## Database



```
8  
9  
10 devise :database_authenticatable, :validatable  
11       :validatable => :password  
12 validates :email, presence: true, uniqueness: true  
13 before_validation :ensure_email_exists  
14 after_initialize :ensure_email_exists  
15 has_many :photos, dependent: :destroy  
16 has_many :orders_placed, dependent: :destroy  
17 has_many :orders_served, dependent: :destroy  
18 has_many :ratings, foreign_key: :user_id  
19 has_many :messages_sent, dependent: :destroy  
20 has_many :messages_received, dependent: :destroy  
21 accepts_nested_attributes_for :photos, :orders_placed, :orders_served  
22  
23 # avatar attachment  
24 # adapter :image_processing => :image_processing  
25
```

2019 onwards

New syllabus

Advanced Level

Sri Lankan

# தரவுத்தள முகாமைத்துவ முறைமை

## தரவுத்தளம் - Database

- தரவுத்தளம் என்பது ஒன்றோடு ஒன்று தொடர்புபட்ட ஒரு தொகுதி தரவுகளின் ஓர் ஒழுங்கமைப்பாகும்.
- தரவுத்தளமானது கை முறை/இலக்க தரவுத் தளமாக காணப்படலாம்.

## தரவுத்தள முகாமைத்துவ முறைமை (Data base management system)

- தரவுத்தளம் ஒன்றினைப் படைத்து , பராமரிப்பதற்கு பயனர்களை அனுமதிக்கின்ற ஒரு தொகுதி செய்நிரல்களாகும்.
- ஓர் கட்டமைக்கப்பட்ட வழிமுறையில் தரவுகளைக் களஞ்சியப்படுத்தி அவற்றை மீள பெறுவதற்கு பயனர்களை அனுமதிக்கின்றது.

Example :

- Oracle
- MySQL
- Ms-Access
- PostgreSQL
- Informix
- Cybase
- Base
- Maria DB
- Ms-SQL Server

## தரவு மாதிரியங்கள்(Data models)

- ஓர் தரவுத்தள முறைமையில் தரவானது எவ்வாறு களஞ்சியப்படுத்தப்பட்டு ஒழுங்கமைக்கப்பட்டு கையாளப்படுகிறது என்பதை தரவு மாதிரியங்கள் தீர்மானிக்கின்றன.
  1. அதிகார படிநிலை மாதிரியம் (Hierarchical model)
  2. வலையமைப்பு நிலை மாதிரியம் (Network Model)
  3. தொடர்புநிலை மாதிரியம் (Relational Model)
  4. உரு தொடர்பு நிலை மாதிரியம் (Object-Relational Model)
  5. தட்டைசோப்பு தரவுத்தள மாதிரியம்

## தொடர்புநிலை மாதிரியம் (Relational model)

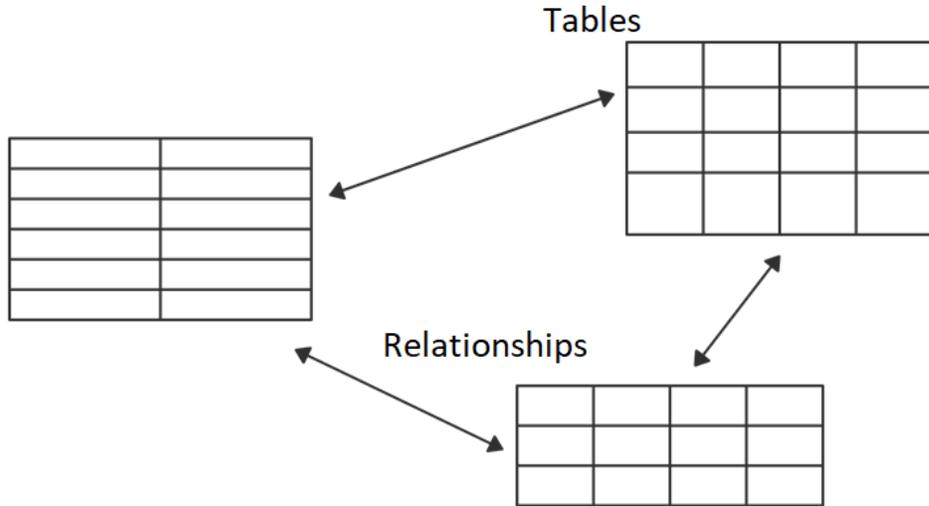
- இது ஒரு பிரபல்யம் மிக்க தரவுத்தளமாகும்.
- பெரும்பாலான தரவுத்தள முகாமைத்துவ முறைமை மென் பொருட்கள் (DBMS) இம் மாதிரியத்தினையே பயன்படுத்துகின்றன.

Edgor Fcodd –Father of relational model

## எண்ணக்கருக்கள் (Concept Map)

- ஓர் தரவுத்தளம் ஒன்று அதற்கு மேற்பட்ட தரவு அட்டவணைகளைக் கொண்டுள்ளது.

**Note:** அட்டவணை / தொடர்பு / உள்பொருள்



## தொடர்பு - Relation

- தொடர்பு என்பது நிரல்கள் மற்றும் நிரைகளைக் கொண்டமைந்த ஓர் அட்டவணையாகும்.

புலங்கள் / பண்புகள்  
Fields/Attributes

Admno	First Name	Last Name	Address	Sex
001	Vimal	Kamal	Jaffna	Male
005	Kajan	Bala	Colombo	Male
006	Arunja	Siva	Kandy	Female

பதிவுகள்  
Records/Tuples

### ❖ தொடர்பினது உடைமைகள்

- ஒவ்வொரு பதிவும் தனித்துவமானது.
- பண்புகள்/புலங்களுக்கு எந்தவொரு வரிசையும் காணப்படமாட்டாது.
- பதிவுகளுக்கு எந்தவொரு வரிசையும் காணப்படமாட்டாது.
- தரவுத்தளமொன்றில் காணப்படுகின்ற ஒவ்வொரு தொடர்பும் ஒன்றில்லொன்று சாராதவை. அவை தம்மிடையே தொடர்புடைமையினை ஏற்படுத்தும்.
- ஒவ்வொரு தொடர்பும் ஒரு தனித்துவமான பெயரினை கொண்டிருத்தல் வேண்டும்.
- தொடர்பானது ஒவ்வொரு கலமும் (Cell) சரியாக ஓர் தனித்த பெறுமதியினைக் (Atomic/Single value) கொண்டிருத்தல் வேண்டும்.
- ஒவ்வொரு பண்பும் தனித்துவமான பெயரினைக் கொண்டிருக்க வேண்டும்.
- ஒவ்வொரு நிரல்களின் பெறுமதிகள் ஒரு குறிப்பிட்ட ஆள்களத்தில் இருத்தல் வேண்டும்.

### **Note:**

- தொடர்பொன்றினது **அளவு/தரம்** என்பது அது கொண்டிருக்கின்ற பண்புகளின் (புலம்) எண்ணிக்கை ஆகும்.
- ஆள்களம் (Domain) - பண்பொன்றிற்கு அனுமதிக்கப்படக் கூடிய பெறுமதிகளின் தொடை ஆள்களமாகும்.(ஒரு அட்டவணையின் ஒரு பண்பிற்கு கீழ் வரும் பெறுமதி ஒரே வகையாக இருத்தல்)

## தொடர்புநிலை தரவுத்தள சாவிகள் (Relational database keys)

1. வேட்பாளர் சாவி/தகுதிகாண் சாவி(candidate key)
2. முதன்மைச்சாவி/பிரதான சாவி (primary key)
3. மாற்றுச்சாவி (Alternate Key)
4. அன்னியச்சாவி/வெளியிடச்சாவி (Foreign Key)
5. ஒருங்குசேர் சாவி/நேர்மான சாவி (Composite Key)

### வேட்பாளர் சாவி – Candidate Key

Cus_ID	Acc_No	NIC	Acc_Type

- தொடர்பு நிலை தரவுத்தளம் ஒன்றில் அட்டவணை ஒன்றின் பதிவுகளை தனித்துவமாக அடையாளப்படுத்துகின்ற அனைத்து புலங்களும் அவ்வட்டவணையின் வேட்பாளர் சாவிகளாகும். (ஒருபுலம்/சில புலங்களின் சேர்க்கை)
- அட்டவணை ஒன்று /அதற்கு மேற்பட்ட வேட்பாளர் சாவிகளைக் கொண்டிருக்க முடியும்.
- இவ்வட்டவணையில் (Cus\_ID, Acc\_No, NIC ) ஆகிய 3 புலங்களும் வேட்பாளர் சாவிகளாக இருக்க முடியும்.

### முதன்மைச்சாவி / பிரதான சாவி – Primary Key

- தொடர்புநிலை தரவுத்தளமொன்றில் தரவட்டவணை ஒன்றின் பதிவுகளை தனித்துவமாக அடையாளப்படுத்துகின்ற புலம் / புலங்களின் தொகுதி அத்தரவட்டவணையின் முதன்மை சாவி என அழைக்கப்படும்.
- முதன்மை சாவி புல பெறுமதிகள் வெற்றுப்பெறுமதிகளாக (Null Values) இருக்க முடியாது.
- தரவுத்தள அட்டவணையில் நிரலை நகல் செய்வதற்கு பயனருக்கு இடமளிக்காது.
- அட்டவணை ஒன்று ஆகக்குறைந்தது ஒரு முதன்மை சாவியை கொண்டிருக்கலாம்.
- அட்டவணை ஒன்று சாவியினை கொண்டிருக்க வேண்டிய அவசியமில்லை(0).
- முதன்மை சாவி புல பெறுமதிகள் மீள்வரும் பெறுமதிகளை (Duplicate Values) கொண்டிருக்க முடியாது.

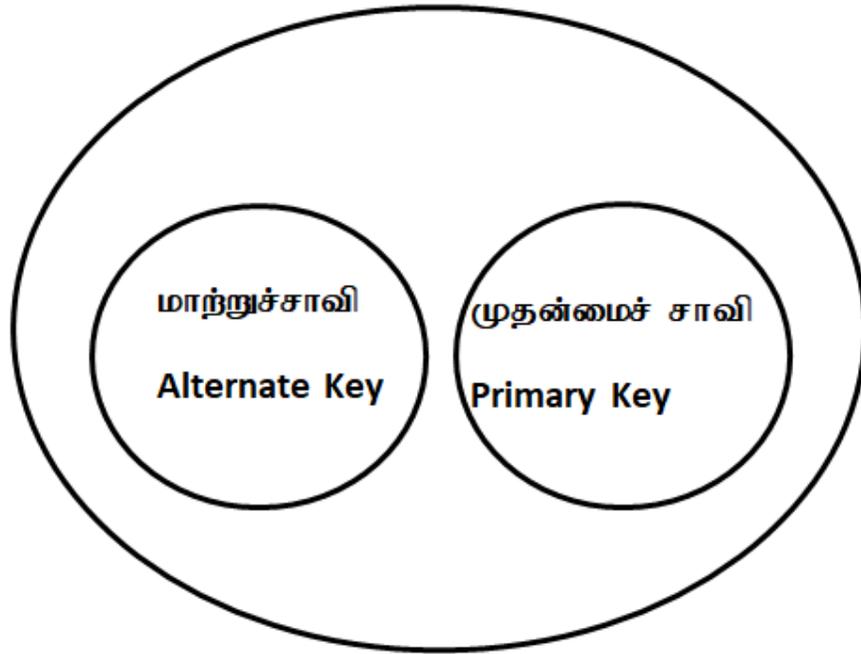
Eg: Student(Admno,Fname,Lname,DOB,Address)

Results (IndexNo,Year,Results)

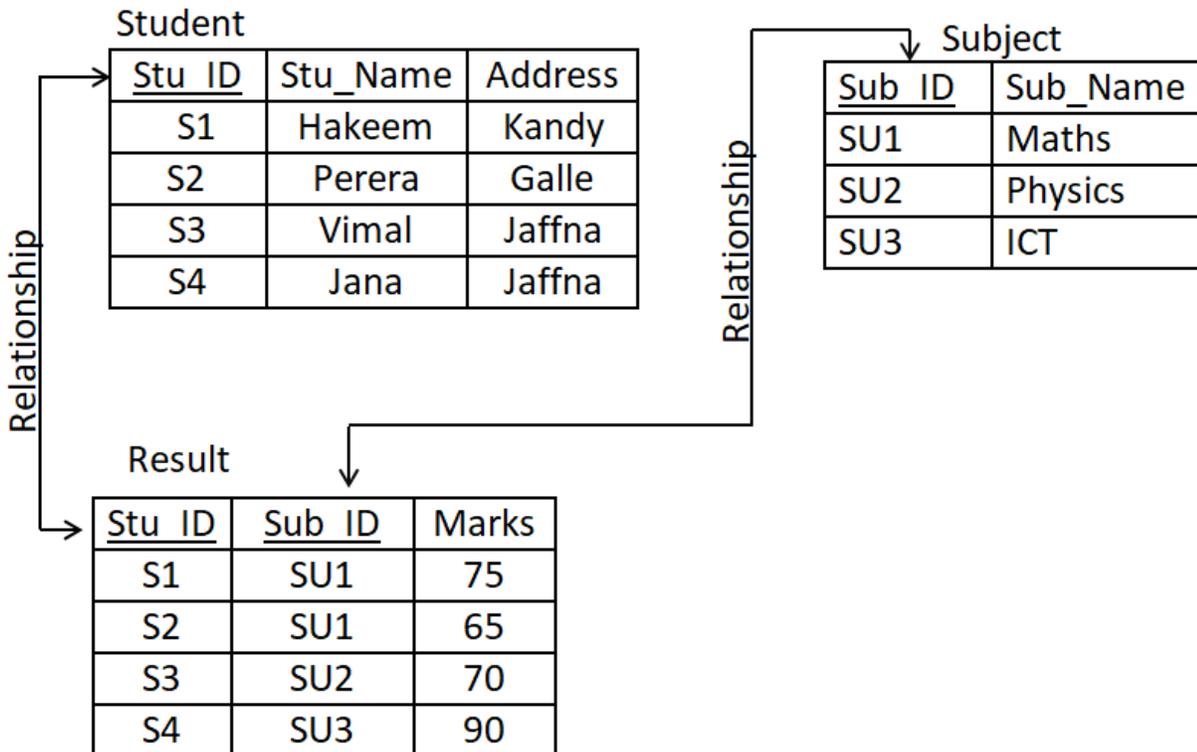
### மாற்றுச் சாவி – Alternate Key

- தொடர்புநிலை தரவுத்தளமொன்றில் முதன்மை சாவியாக அட்டவணை ஒன்றில் தெரிவு செய்யப்பட்ட புலத்தினை தவிர ஏனைய வேட்பாளர் சாவிகள் அனைத்தும் மாற்றுச்சாவிகளாகும்.
  - மேலுள்ள அட்டவணையில் CusID முதன்மை சாவியாக தெரிவு செய்யப்பட்டின் AccNo,NIC ஆகிய இரண்டு புலங்களும் அவ்வட்டவணையில் மாற்றுச்சாவிகளாகும்.

## Candidate key - வேட்பாளர் சாவி



### அன்னியச்சாவி / வெளியிடச்சாவி - Foreign Key



- தொடர்புநிலை தரவுத்தளமொன்றின் அந்நியச்சாவி என்பது ஓர் அட்டவணையின் புலம் / புலங்களின் சேர்க்கை ஆகும். அது இன்னொரு அட்டவணையின் ஒரே முதன்மை சாவி புல பெறுமதிகளை மேற்கோள் இடுகின்றது.
- தொடர்புநிலை தரவுத்தளம் ஒன்றில் அட்டவணை ஒன்றின் முதன்மை சாவியும் இன்னொரு அட்டவணையின் அந்நிய சாவியும் சேர்ந்து அவ்விரண்டு அட்டவணைக்களுக்கிடையே தொடர்புடைமையினை (**Relationship**) ஏற்படுத்துகின்றன.

ஒருங்குசேர் சாவி - Composite key

Supplier_ID	Parts_ID	Quantity
S1	P1	200
S1	P2	300
S2	P1	400
S3	P1	150
S3	P2	700

- தொடர்புநிலை தரவுத்தளமொன்றில் தரவட்டவணை ஒன்றின் பதிவுகளை தனித்துவமாக அடையாளப்படுத்துகின்ற இரண்டு / அதற்கு மேற்பட்ட புலங்களின் சேர்க்கை அவ்வட்டவணையின் ஒருங்குசேர் சாவி ஆகும்.

Supplier\_ID+Parts\_ID=Composite Key

- ஒருங்குசேர் சாவி முதன்மை சாவிடாகவும் தொழிற்படலாம் ஆனால் முதன்மை சாவி ஒருங்குசேர் சாவிடாக தொழிற்படாது.

## ER-Diagram - Entity Relationship Diagram

### உள்பொருள் அட்டவணை - Entity



- உள்பொருள் என்பது நடைமுறை உலகில் காணப்படக்கூடிய ஓர் சுதந்திரமான இருப்புடைய ஓர் பொருளாகும்.

Eg: Person, Car, House, Employee, Subject, Student

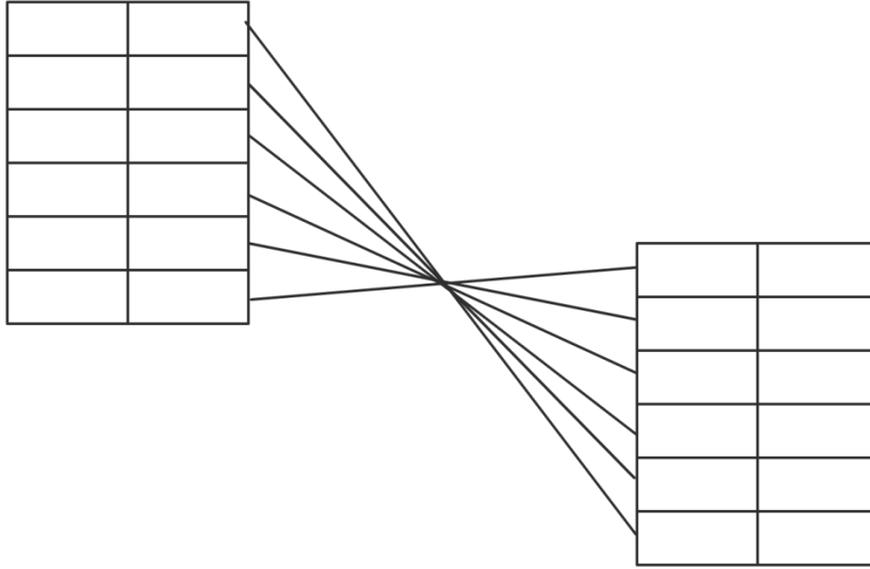
### ❖ உள்பொருளின் வகைகள் (Types of Entity)

1. வலுவான உள்பொருள் (Strong Entity )
2. பலவீனமான உள்பொருள்(weak Entity)

### தொடர்புடைமை (Relationship)

- இரண்டு அட்டவணைக்களுக்கிடையிலான ஓர் கருத்துள்ள இணைவு தொடர்புடைமை என அழைக்கப்படும்.



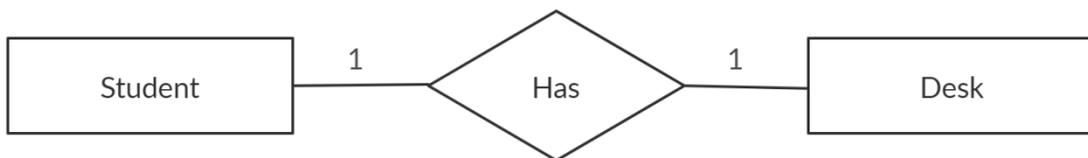


## தொடர்புடைமையின் வகைகள் (Types of Relationship)

1. ஒன்றுக்கொன்றான தொடர்புடைமை(1:1)
- 2.ஒன்றுக்கு பலவான தொடர்புடைமை(1:m or 1:n)
3. பலவுக்கு பலவான தொடர்புடைமை(m:n)

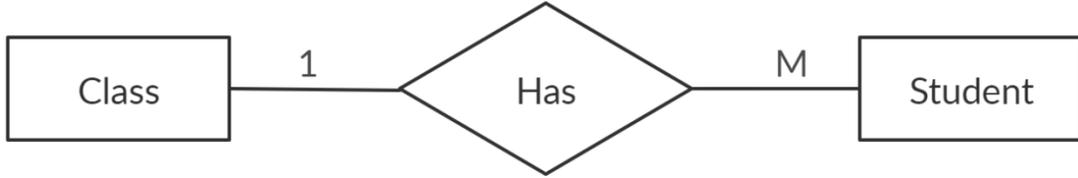
### ❖ ஒன்றுக்கொன்றான தொடர்புடைமை (1:1)

- ஒரு Desk ஒரு மாணவனுக்கு உரித்துடையதாக இருக்கின்ற அதேவேளை ஒரு மாணவன் ஒரு Desk இனை கொண்டிருப்பார்.



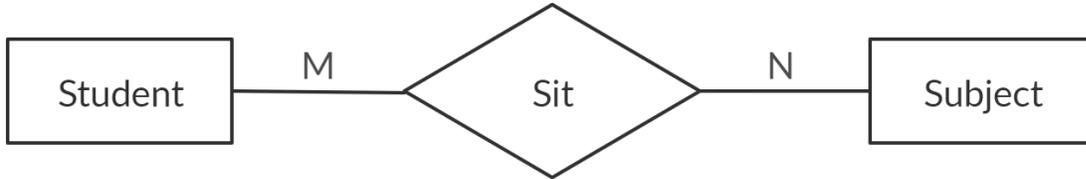
### ❖ ஒன்றுக்கு பலவான தொடர்புடைமை(1:m or 1:n)

- ஒரு Class பல மாணவர்களை கொண்டிருக்கின்ற அதேவேளை ஒரு மாணவன் ஒரு மாணவன் ஒரு Class ற்கு உரித்துடையவர்.



### • பலவுக்கு பலவான தொடர்புடைமை (m:n)

- ஒரு மாணவன் ஒரு பாடங்களுக்கு தோற்றுக்கின்ற அதே வேளை ஒரு பாடம் ஒன்றுக்கு மேற்பட்ட மாணவர்களினால் எடுக்கப்படுகின்றது.



### தொடர்புடைமை அளவு/தரம் (Degree)

- தொடர்புடைமை ஒன்றில் பங்கு பற்றுக்கின்ற உள்பொருள்களின் எண்ணிக்கை அத்தொடர்புடைமையானது அளவு ஆகும்.

Degree(2) - Binary

Degree(3) - Ternary

Degree(4) - Quaternary

## பண்பு/புலம் - Attribute

- உள்பொருள்/ தொடர்புடைமை ஒன்றினது உடைமை பண்பு ஆகும்.

AdmNumber

- பண்பினது வகைகள் - Types of Attribute

### 1.எளிய பண்பு (simple attribute)

- மேலும் சிறிய கூறுகளாக பிரிக்கப்பட முடியாத பண்பு இதுவாகும்.

Eg:EmpNo

Emp\_No

### 2.பலபெறுமதிப்பண்பு (Multi valued attribute)

- பண்பொன்று ஒன்றிற்கு மேற்பட்ட பெறுமதிகளை எடுக்கின்ற சந்தர்ப்பம்.

- Eg:- Employee skill

Qualification

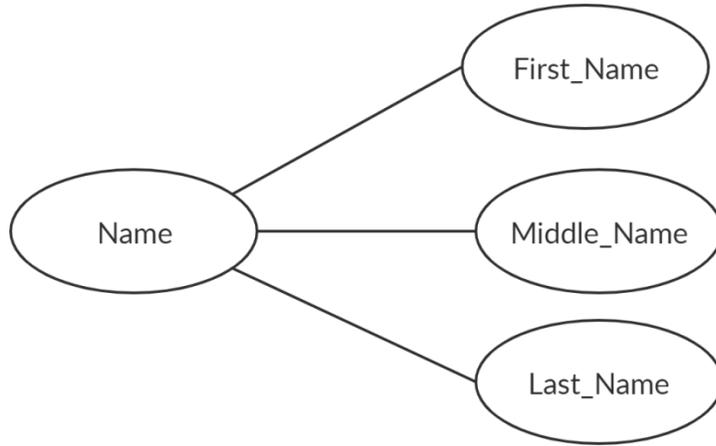
Skill

### 3.ஒருங்குசேர் பண்பு - Composite Attribute

- பண்பொன்று மேலும் பிரிக்கப்படக்கூடிய கூறுகளைக் கொண்டிருத்தல்.

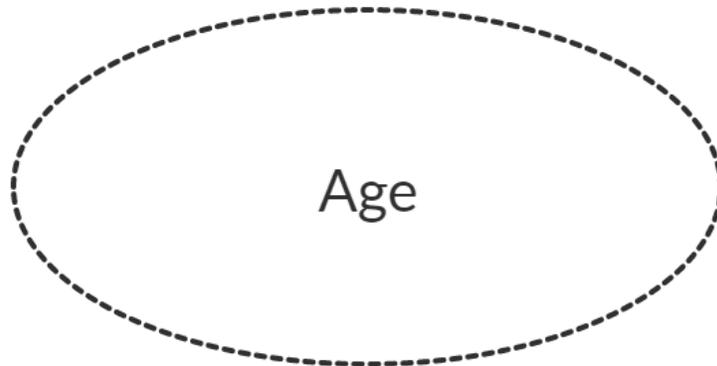
Eg:- Name

Address



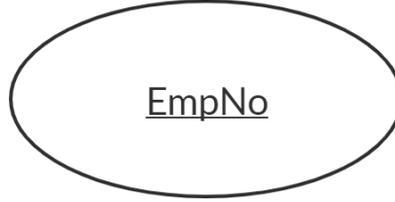
### 4.பெற்ற பண்பு - Derived Attribute

- பண்பு ஒன்றின் பெறுமதிகளில் இருந்து இன்னொரு பண்பின் பெறுமதிகள் கணிக்கபடுதல்.



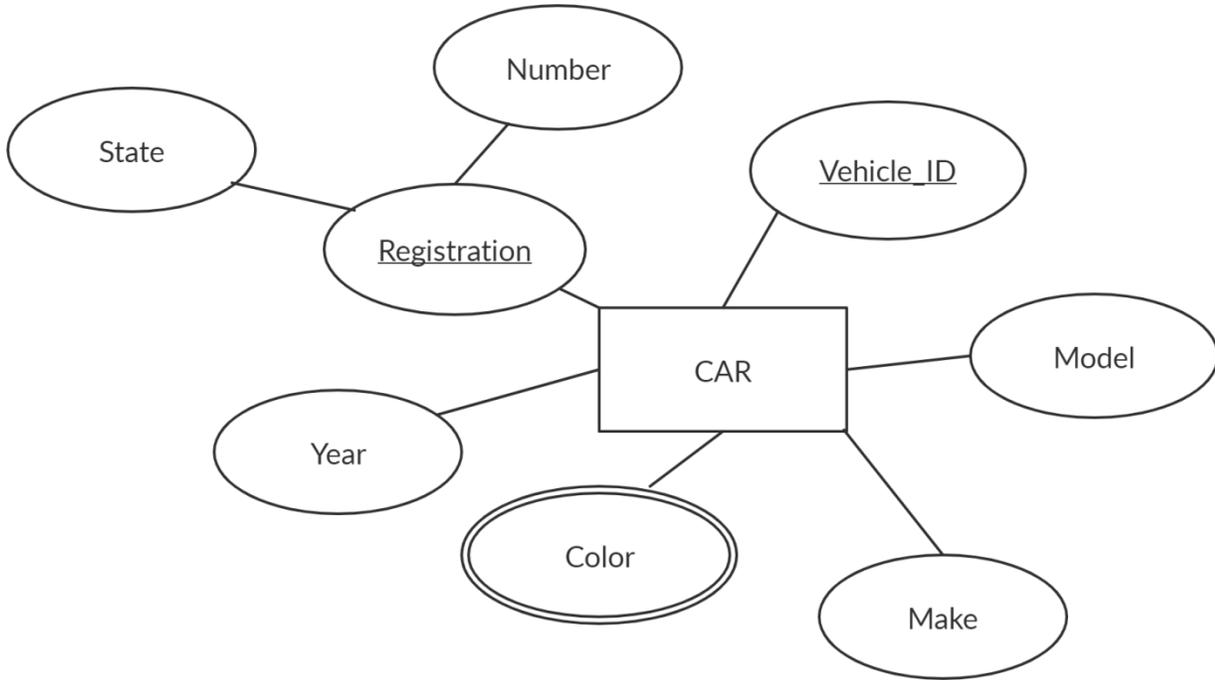
## 5. அடையாளப்படுத்தும் பண்பு - Identifier Attribute

- பதிவுகளை தனித்துவமாக அடையாளப்படுத்துகின்ற பண்பு இதுவாகும்



### Examples

01.

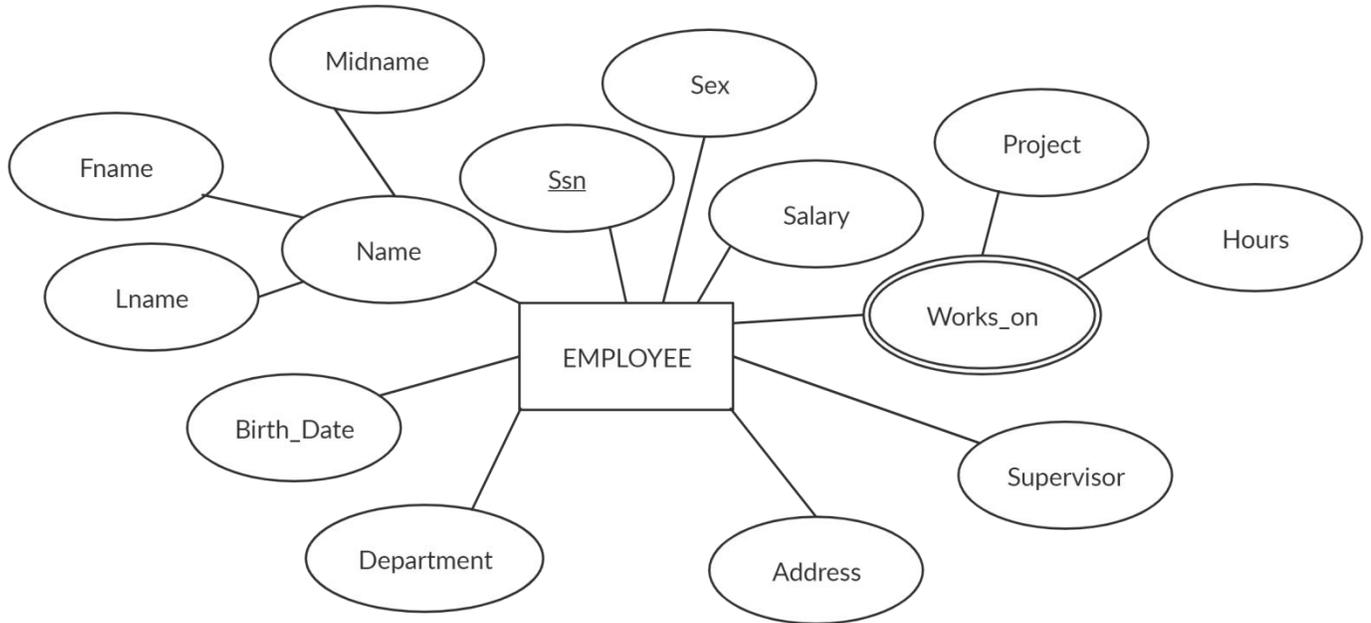


Write down types of Attributes.

- I. Color - பல்பெறுமதிப்பண்பு
- II. Vehicle - அடையாளப்படுத்தும் பண்பு
- III. Registration - ஒருங்குசேர் அடையாளப்படுத்தும் பண்பு

CAR(Make, Model, Vehicle\_ID, Year, State, Number, Color)

2.



**Ssn** - அடையாளப்படுத்தும் பண்பு

**Works\_on** - பல்பெறுமதிப்பண்பு

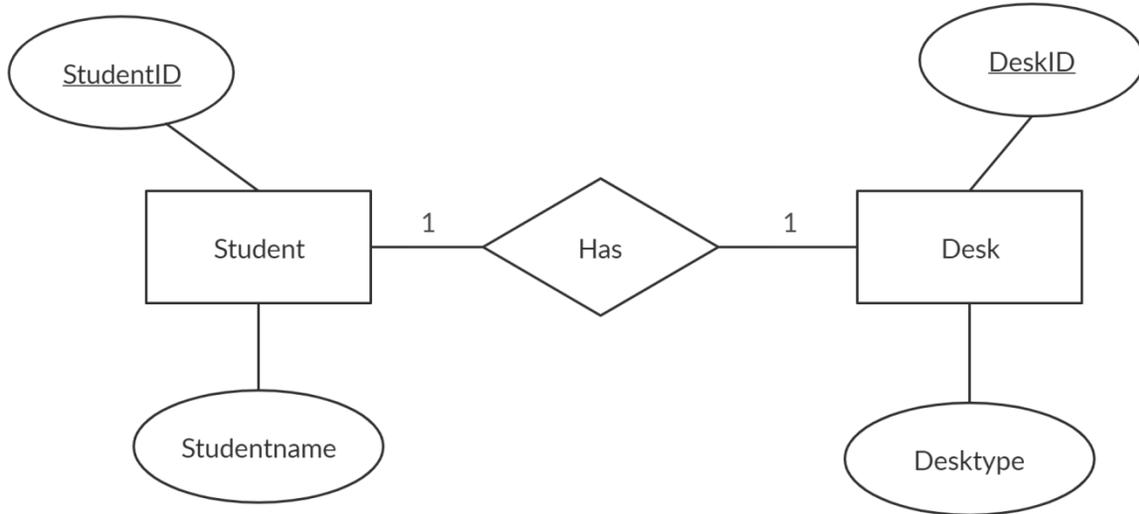
**Name** - ஒருங்குசேர் பண்பு

EMPLOYEE(Ssn, Sex, Salary, Department, Birth\_Date, Address, Supervisor, Fname, Midname, Lname, Project, Hours)

## Mapping / converting ER-Diagram into relations

ER-Diagram ஐ அட்டவணைகளுக்கு படமிடல் / மாற்றுதல்

### ❖ One to one



தீர்வு 1

Student(StudentID, Studentname, DeskID)

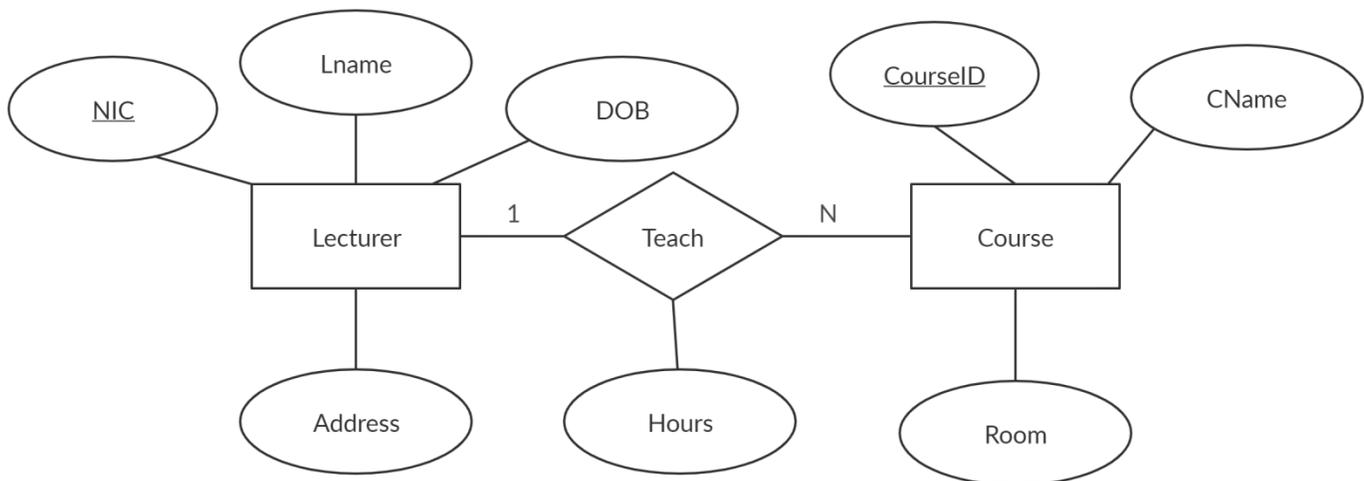
Desk(DeskID, Desktype, )

தீர்வு 2

Student(StudentID, Studentname)

Desk(DeskID, Desktype, StudentID)

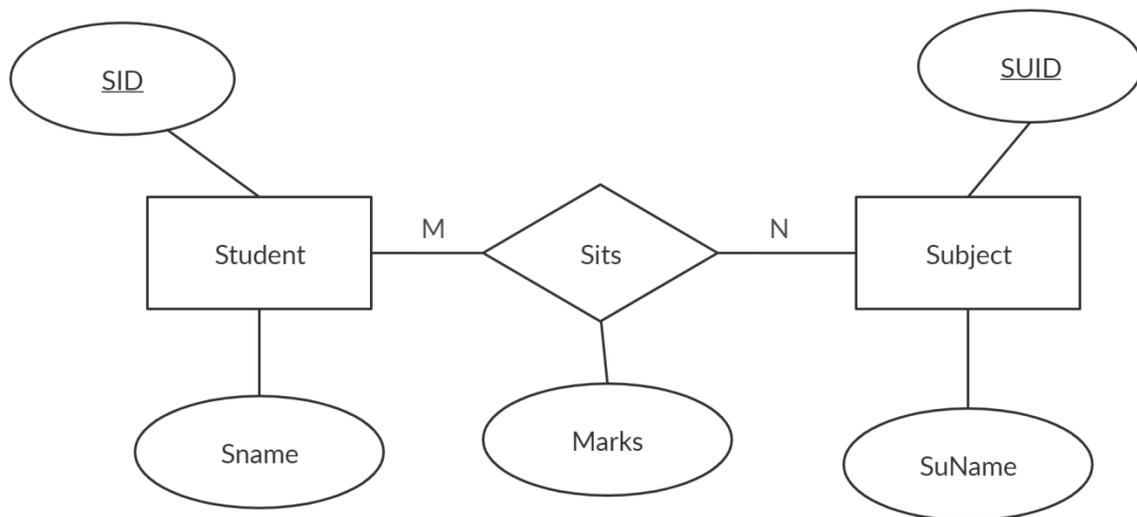
## ❖ One to many



Lecturer(NIC, Lname, DOB, Address)

Course(CourseID, CName, Room, Hours)

## ❖ many to many



Student(SID, Sname)

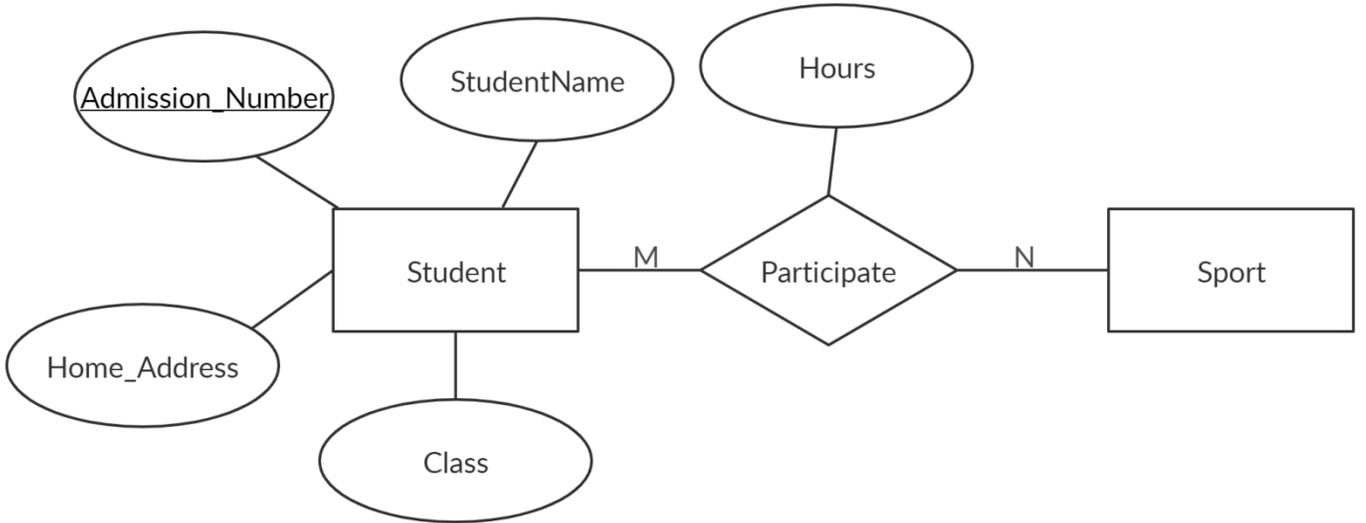
Subject(SUID, SuName)

Student\_Subject(SID, SUID, Marks)

01.பின்வரும் நிலைமையினைக் கருதுக.

பாடசாலை ஒன்றில் மாணவர்கள் வேறுபட்ட விளையாட்டு நிகழ்வுகளான (volley ball, track and field athletics, Table tennis ) போன்றவற்றில் பங்கு கொள்கின்றனர். பாடசாலையின் அதிபர் (admission number, student name ,Home address, class) போன்ற விபரங்களை பராமரிப்பதுடன் அவர் பங்கு பற்றுக்கின்ற விளையாட்டு தொடர்பிலும் தரவுகளை பேணுகின்றார். ஒரு மாணவன் ஒன்றிற்கு மேற்பட்ட விளையாட்டுக்களில் பங்குபற்ற முடிவதுடன் ஒரு விளையாட்டில் ஒன்றிற்கு மேற்பட்ட மாணவர்கள் இருக்க முடியும். ஒவ்வொரு மாணவனும் ஒரு விளையாட்டில் முன்னேற தீர்மானிக்கப்பட்ட மணித்தியாலங்கள் பங்குபற்றுவார்.

- இந்தநிலைமைகளை (ER) வரிப்படத்தை வரைக.



- மேலே (1)ல் அடையாளப்படுத்தப்பட்ட தொடர்புடைமையின் முதலிமை(Cardinality) ஒன்றுக்கொன்றா /ஒன்றுக்கு பலவா /பலவுக்கு பலவா எனக்காரணத்துடன் குறிப்பிடுக  
பலவுக்கு பலவென (Many to Many) தொடர்புடைமை

- '(ER) வரிப்படங்கள் தொடர்புடைமையின் மீது பண்புகளை அனுமதிப்பதில்லை' இக்கூற்று உண்மையானதா பொய்யானதா தரப்பட்ட நிலைமையினை பயன்படுத்தி விடையை விளக்குக?
  - பொய்  
(Hours) எனும் பண்பு (Participate) தொடர்புடைமையின் மீது ஓர் பங்கீட்டு பண்பாக குறிக்கப்படல் வேண்டும்.
- தரவுத்தள வடிவமைப்பாளரொருவர் மேலுள்ள முறைமைக்காக பின்வரும் அட்டவணையை பிரேரிக்கின்றார் இவ்வட்டவணையின் இரண்டு குறைபாட்டை குறிப்பிட்டு தேவையான மாற்றங்களை பிரேரிக்குக
  - தரவு வேள் மிகை / தரவு மறுபதிவு

(Data Redundancy/Data duplication)

(Hours) எனும் பண்பு கருத்தில் கொள்ளப்படவில்லை.

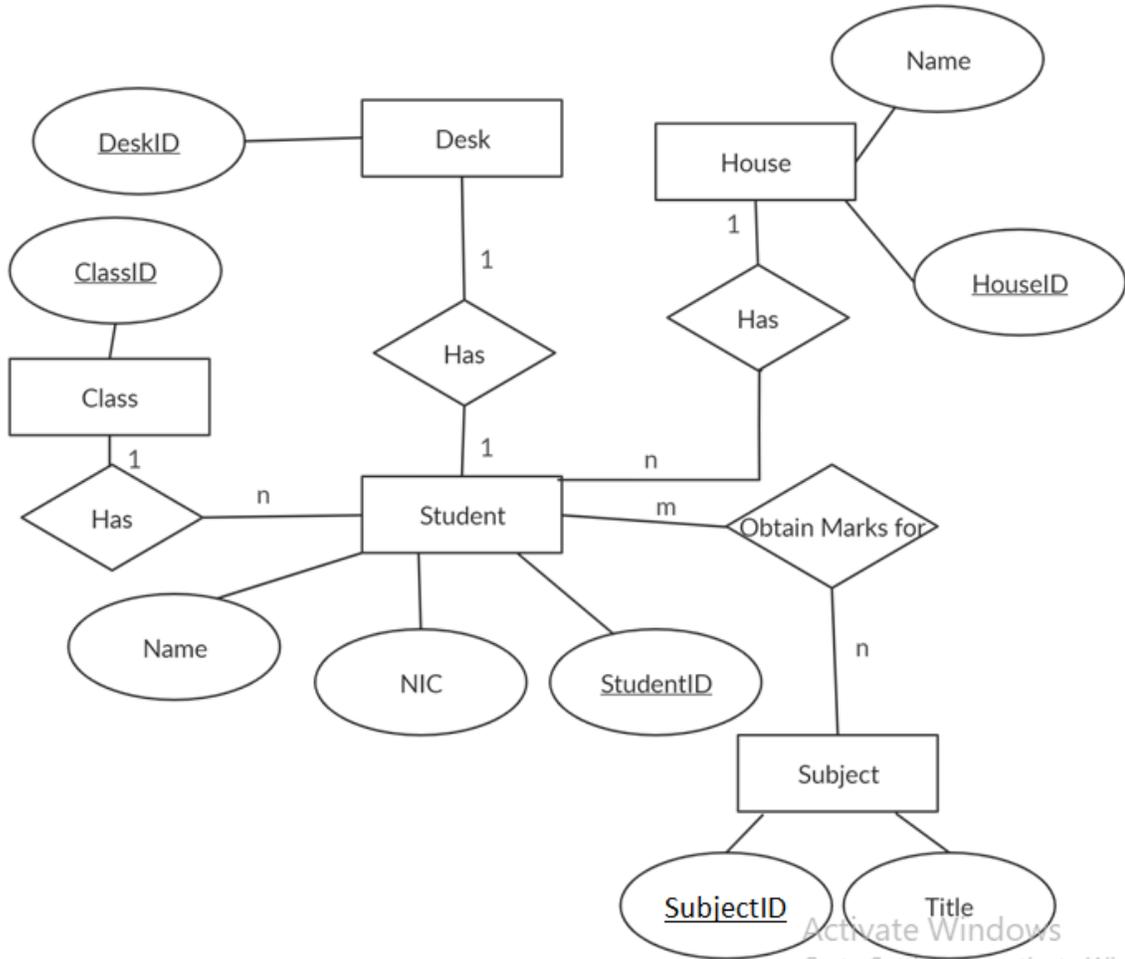
Student(Admission number, StudentName, Home\_Address, Class)

Sport(SportID, Sportname)

Student\_Sport(Admission number, SportID, Hours)

2012

பின்வரும் ER வரிப்படத்தை கருதுக.



- I. மாணவரொருவரினால் பாடமொன்றிற்கு பெற்ற புள்ளிகளை வகைகுறிப்பதற்கு ER - வரிப்படத்தில் பொருத்தமான உட்பொருள் / தொடர்புடைமையை குறிப்பிடுக.
- Obtain Marks for எனும் தொடர்புடைமை மீது குறிக்கப்பட வேண்டும்

- II. ER - வரிப்படத்தில் அனைத்து தொடர்புடைமைகளையும் ஒன்றுக்கொன்றா /ஒன்றுக்கு பலவா /பலவுக்கு பலவா எனக்காரணத்துடன் குறிப்பிடுக
- Class-Has-Student : One to many
    - Class ஒன்று பல Studentஐ கொண்டிருக்கும் அதேவேளை ஒரு Classற்கு ஒரு Student உரித்துடையவர்.
  - Student-Has-Desk : One to one
    - ஒரு Desk ஒரு Studentற்கு உரித்தாகும் அதேவேளை ஒரு Student ஒரு Desk கொண்டிருப்பார்.
  - House-Has-Student : One to Many
    - ஒரு Student ஒரு Houseற்கு உரித்துடையதாக இருக்கும் அதேவேளை ஒரு House பல மாணவர்களைக் கொண்டிருக்கும்.
  - Student-Obtains marks for -Subject : many to many
    - ஒரு Student ஒன்றிற்கு மேற்பட்ட Subjectற்கு தோற்றும் அதேவேளை ஒரு Subject ஒன்றுக்கு மேற்பட்ட Student இனால் எடுக்கப்படலாம்.
- III. ER - வரிப்படம் தொடர்புநிலை தரவுத்தளத்திலுள்ள அட்டவணைகளுக்கு படமிடப்பட்டால் அட்டவணைகளின் கட்டமைப்பினை அவற்றின் பண்புகளுடன் எழுதுக.
- Student(StudentID, Name, NIC, ClassID, HouseID, DeskID)
  - Subject(SubjectID, Title)
  - Student\_Subject(StudentID, SubjectID, Marks)
  - Class(ClassID)
  - Desk(DeskID)
  - House(HouseID, Name)

IV. மேலே குறிப்பிடப்பட்ட அட்டவணைகள் ஒவ்வொன்றுக்கும் பொருத்தமான முதன்மைச்சாவினைக் குறிப்பிடுக.

- Student - StudentID
- Subject - SubjectID
- Student\_Subject - StudentID + SubjectID
- Class - ClassID
- Desk - DeskID
- House - HouseID

V. பெறப்பட்ட அட்டவணைகளை பயன்படுத்தி StudentID ஆனது ST001 ஆகவுள்ள மாணவரொருவரினால் SubjectID AL001 ஆகவுள்ள பாடத்திற்கு பெற்ற புள்ளிகளை பெறுவதற்கான SQL கூற்றை எழுதுக.

```
select marks
from Student_Subject
where StudentID='ST001' AND SubjectID='AL001';
```

# SQL (Structured Query Language)

- MySQL can be run in these following Servers;
  - 1.app sever
  - 2.WAMP server
  - 3.XAMPP server
- Here we are Using Appserv
  - How to install Appserv Properly with Password:-  
<https://www.appserv.org/en/howto/>
- Running MySql in Command Prompt
  - (Enter the Password which you have given during the installation)
  - `mysql -u root -p`

`mysql<space>-u<space>root<space>-p)`

```
C:\Users\hp>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- Now we can Start Mysql Coding...

## NOTE

**The things  
displaying after ;  
are the outputs**

**\*All commands are used in small  
Alphabets**

**\*No need to write mysql> in Exam  
papers**

**Create Database - தரவுத்தளமொன்றை உருவாக்குதல்**

```
mysql> create database employees;  
Query OK, 1 row affected (1.49 sec)
```

**Show Databases - தரவுத்தளங்களை காட்சிப்படுத்தல்**

```
mysql> show databases;
+-----+
| Database |
+-----+
| employees |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.43 sec)
```

**Drop Database - தரவுத்தளமொன்றை அழித்தல்**

```
mysql> create database Hello;
Query OK, 1 row affected (0.15 sec)

mysql> drop database Hello;
Query OK, 0 rows affected (0.95 sec)
```

Use Database - தரவுத்தளமொன்றிற்குள் செல்லுதல்

```
mysql> use employees;  
Database changed
```

Create Table - அட்டவணை ஒன்றை உருவாக்கல்

```
mysql> create table employee(empid varchar(7)  
primary key,name varchar(25),address varchar  
(50),salary int(8));  
Query OK, 0 rows affected, 1 warning (1.93 se  
c)
```

Drop Table - அட்டவணையொன்றை அழித்தல்

```
mysql> drop table employee;  
Query OK, 0 rows affected (0.76 sec)
```

**Show Tables - அட்டவணைகளை காட்சிப்படுத்தல்**

```
mysql> show tables;
+-----+
| Tables_in_employees |
+-----+
| employee             |
+-----+
1 row in set (0.19 sec)
```

**Describe Table - அட்டவணையொன்றை விபரித்தல்****Method 1**

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empid | varchar(7)    | NO   | PRI | NULL    |      |
| name  | varchar(25)   | YES  |     | NULL    |      |
| address | varchar(30)  | YES  |     | NULL    |      |
| salary | int(8)        | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.07 sec)
```

**Method 2**

```
mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empid      | varchar(7)    | NO   | PRI | NULL    |       |
| name       | varchar(25)   | YES  |     | NULL    |       |
| address    | varchar(30)   | YES  |     | NULL    |       |
| salary     | int(8)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.17 sec)
```

**Insert Record - பதிவுகளை உள்ளுளைத்தல்**

```
mysql> insert into employee values('e01','Amal','Jaffna',50000);
Query OK, 1 row affected (0.25 sec)

mysql> insert into employee values('e02','Bala','Colombo',35000);
Query OK, 1 row affected (0.14 sec)

mysql> insert into employee values('e03','Sutha','Kandy',35000);
Query OK, 1 row affected (0.18 sec)

mysql> insert into employee values('e04','Thaksha','Jaffna',40000);
Query OK, 1 row affected (0.09 sec)

mysql> insert into employee values('e05','Angel','Galle',50000);
Query OK, 1 row affected (0.10 sec)

mysql> insert into employee values('e06','Kavya','Vavuniya',45000);
Query OK, 1 row affected (0.07 sec)
```

**Select**

- அனைத்து பதிவுகளையும் பார்த்தல்/பதிவு செய்தல்

```
mysql> select *  
-> from employee;
```

empid	name	address	salary
e01	Amal	Jaffna	50000
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000
e05	Angel	Galle	50000
e06	Kavya	Vavuniya	45000

```
6 rows in set (0.00 sec)
```

❖ **salary** மற்றும் **empid** ஐ காட்சிப்படுத்தல்

```
mysql> select empid,salary
-> from employee;
+-----+-----+
| empid | salary |
+-----+-----+
| e01   | 50000  |
| e02   | 35000  |
| e03   | 35000  |
| e04   | 40000  |
| e05   | 50000  |
| e06   | 45000  |
+-----+-----+
6 rows in set (0.00 sec)
```

❖ **salary** ஆனது 38000 ஐ விட அதிகமாக இருக்கும் அனைத்து பதிவுகளையும் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> where salary>38000;
+-----+-----+-----+-----+
| empid | name   | address | salary |
+-----+-----+-----+-----+
| e01   | Amaal | Jaffna  | 50000 |
| e04   | Thaksha | Jaffna  | 40000 |
| e05   | Angel  | Galle   | 50000 |
| e06   | Kavya  | Vavuniya | 45000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

❖ **address 'Jaffna'** ஆக உள்ள அனைத்து பதிவுகளையும் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> where address='Jaffna';
+-----+-----+-----+-----+
| empid | name   | address | salary |
+-----+-----+-----+-----+
| e01   | Amaal | Jaffna  | 50000 |
| e04   | Thaksha | Jaffna  | 40000 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

❖ **address 'Jaffna'** ஆக இல்லாத அனைத்து பதிவுகளையும் காட்சிப்படுத்துக.

Method 1

```
mysql> select *
-> from employee
-> where address<>'Jaffna';
```

empid	name	address	salary
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e05	Angel	Galle	50000
e06	Kavya	Vavuniya	45000

4 rows in set (0.00 sec)

Method 2

```
mysql> select *
-> from employee
-> where not address='Jaffna';
```

empid	name	address	salary
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e05	Angel	Galle	50000
e06	Kavya	Vavuniya	45000

4 rows in set (0.03 sec)

❖ **address 'Jaffna'** ஆக அல்லது **salary 37000** ற்கு அதிகமாக அல்லது சமனாக உள்ள அனைத்து பதிவுகளையும் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> where address='Jaffna' or salary>=37000;
```

empid	name	address	salary
e01	Amal	Jaffna	50000
e04	Thaksha	Jaffna	40000
e05	Angel	Galle	50000
e06	Kavya	Vavuniya	45000

4 rows in set (0.00 sec)

❖ **address 'Jaffna'** ஆகவும் **salary 40000** ற்கு சமனாகவும் உள்ள அனைத்து பதிவுகளையும் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> where address='Jaffna' and salary=40000;
```

empid	name	address	salary
e04	Thaksha	Jaffna	40000

1 row in set (0.00 sec)

- ❖ **employee** அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் **salary** இன் இறங்குவரிசையில் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> order by salary desc;
```

empid	name	address	salary
e01	Amal	Jaffna	50000
e05	Angel	Galle	50000
e06	Kavya	Vavuniya	45000
e04	Thaksha	Jaffna	40000
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000

6 rows in set (0.00 sec)

- ❖ **order by salary** என்று **Query** கொடுக்கப்படும் போது **Default**

ஆகவே ஏறுவரிசையில் **Output** காட்சிப்படுத்தப்படும்

```
mysql> select *
-> from employee
-> order by salary;
```

empid	name	address	salary
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000
e06	Kavya	Vavuniya	45000
e01	Amal	Jaffna	50000
e05	Angel	Galle	50000

6 rows in set (0.00 sec)

- ❖ **employee** அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் **salary** இன் ஏறுவரிசையில் காட்சிப்படுத்துக.

```
mysql> select *
-> from employee
-> order by salary asc;
```

empid	name	address	salary
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000
e06	Kavya	Vavuniya	45000
e01	Amal	Jaffna	50000
e05	Angel	Galle	50000

```
6 rows in set (0.00 sec)
```

- ❖ **employee** அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் **name** இன் ஏறுவரிசையில் காட்சிப்படுத்துக

```
mysql> select *
-> from employee
-> order by name asc;
```

empid	name	address	salary
e01	Amal	Jaffna	50000
e05	Angel	Galle	50000
e02	Bala	Colombo	35000
e06	Kavya	Vavuniya	45000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000

```
6 rows in set (0.03 sec)
```

like

### 1. names starting with 'A'

```
mysql> select *
-> from employee
-> where name like 'A%';
```

empid	name	address	salary
e01	Amal	Jaffna	50000
e05	Angel	Galle	50000

```
2 rows in set (0.05 sec)
```

### 2. names having letter 'a'

```
mysql> select *
-> from employee
-> where name like '%a';
```

empid	name	address	salary
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000
e06	Kavya	Vavuniya	45000

```
4 rows in set (0.00 sec)
```

### 3. names having letter 's'

```
mysql> select *
-> from employee
-> where name like '%s%';
+-----+-----+-----+-----+
| empid | name   | address | salary |
+-----+-----+-----+-----+
| e03   | Sutha  | Kandy   | 35000  |
| e04   | Thaksha | Jaffna  | 40000  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

### 4. names having letters 'an' continuously

```
mysql> select *
-> from employee
-> where name like 'an%';
+-----+-----+-----+-----+
| empid | name   | address | salary |
+-----+-----+-----+-----+
| e05   | Angel  | Galle   | 50000  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

5.

```
mysql> select *
      -> from employee
      -> where name like 'ad%';
Empty set (0.00 sec)
```

### Sum

1.

```
mysql> select sum(salary)
      -> from employee;
+-----+
| sum(salary) |
+-----+
|      255000 |
+-----+
1 row in set (0.05 sec)
```

2.

```
mysql> select sum(salary) as Total_salary
-> from employee;
+-----+
| Total_salary |
+-----+
|      255000 |
+-----+
1 row in set (0.00 sec)
```

**Average**

```
mysql> select avg(salary) as Average_salary
-> from employee;
+-----+
| Average_salary |
+-----+
|  42500.0000 |
+-----+
1 row in set (0.00 sec)
```

**Maximum**

```
mysql> select max(salary) as Maximum_salary
-> from employee;
+-----+
| Maximum_salary |
+-----+
|          50000 |
+-----+
1 row in set (0.04 sec)
```

**Count**

```
mysql> select count(empid) as No_of_Employees
-> from employee;
+-----+
| No_of_Employees |
+-----+
|                6 |
+-----+
1 row in set (0.22 sec)
```

## Update

## ❖ Update salary

```
mysql> update employee
-> set salary=52000
-> where empid='e05';
Query OK, 1 row affected (0.12 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select *
-> from employee
-> where empid='e05';
```

```
+-----+-----+-----+-----+
| empid | name  | address | salary |
+-----+-----+-----+-----+
| e05   | Angel | Galle   | 52000  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## ❖ Update details of a specific record

```
mysql> update employee
  -> set address='Nallur',salary=50000
  -> where empid='e05';
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

## ❖ பதிவு ஒன்றினை அழித்தல்

```
mysql> delete
  -> from employee
  -> where empid='e06';
Query OK, 1 row affected (0.12 sec)
```

## ❖ புதிதாக புலமொன்றை சேர்த்தல்

```
mysql> alter table employee
  -> add email varchar(20);
Query OK, 0 rows affected (0.69 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

❖ புலம் ஒன்றினை அழித்தல்

```
mysql> alter table employee
-> drop email;
Query OK, 0 rows affected (1.54 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

❖ ஏற்கனவேயுள்ள புலத்தின் பெயரினை மாற்றுதல்

```
mysql> alter table employee
-> change name first_name varchar(30);
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select *
-> from employee
-> ;
```

empid	first_name	address	salary
e01	Amal	Jaffna	50000
e02	Bala	Colombo	35000
e03	Sutha	Kandy	35000
e04	Thaksha	Jaffna	40000
e05	Angel	Nallur	50000

❖ தரவு வகையினை மாற்றியமைத்தல்

### Method 1

```
mysql> alter table employee  
-> modify column salary float(10);  
Query OK, 5 rows affected (1.29 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

### Method 2

```
mysql> alter table employee  
-> modify salary float(10);  
Query OK, 0 rows affected (0.34 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

## தரவுத்தள செவ்வணாக்கம் - Database Normalization

- தரவுத்தள செவ்வணாக்கம் என்பது தரவு மேள்மிகையினை இழிவாக்கும் பொருட்டு தொடர்புநிலை தரவுத்தள ஒன்றில் அட்டவணைகள் மற்றும் புலங்கள் போன்றவற்றை ஒழுங்கமைக்கின்ற செயன்முறை ஆகும் .
- இதன் போது பாரிய தரவுஅட்டவணைகள் சிறிய தரவுஅட்டவணைகளாக பிரிக்கப்பட்டு அவற்றிலிடையே தொடர்புடைமைகள் ஏற்படுத்தப்படும்.

### ❖ நன்மைகள் / அனுகூலங்கள் / நோக்கம்

- அட்டவணை ஒன்றில் இழிவளவான எண்ணிக்கையில் புலங்களை/பண்புகளை பேணல்.
- அட்டவணை ஒன்றில் புலங்களிடையே ஒரு நெருக்கமான தருக்க தொடர்புடைமையை பேணல்.
- தரவு மேள்மிகையினை இழிவு நிலையில் பேணல்.
- தரவு முரண்களை நீக்குதல் (data anomalies)

## ONF Zero normalization

- அட்டவணை ஒன்று ஒன்று/அதற்கு மேற்பட்ட மீள்வரும் குழுக்களாக(Repeating Group) கொண்ட பெறுமதிகளை கொண்டிருக்கும் நிலைமை (செவ்வணாக்கப்படாத வடிவம் -unnormal form)
- ❖ ONF முகம் கொடுக்கும் பிரச்சினைகள்

- வினைத்திறனான முறையில் தரவுகளை கையாள முடியாமை.(அதாவது தகவலை தேடி பெறுவதில் பெறுவதில் உரிய தரவை பெறுவதில் தாமதம்,சிக்கல் )
- சரியான முறையில் ,தேவையான தகவலை பதிதல், மற்றும் பெற்று கொள்வதில் காணப்படக்கூடிய பிரச்சினை.

CusID	CusName	TelNo
C1	Saman	021 222 2332 079 548 7683
C2	Vimal	024 456 7890
C3	Kamal	021 222 9160 077 341 2556 076 876 7896

Repeating Groups

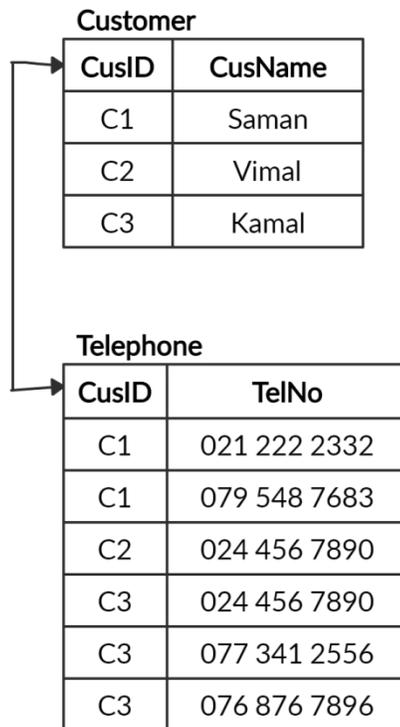
## 1NF [First Normal Form]

- அட்டவணை ஒன்று மீள்வரும் குழுக்களின் கொண்ட பெறுமதியை (repeating groups) கொண்டிருக்காவிடத்து அது 1ம் செவ்வன் வடிவத்தில் இருப்பதாக கூறப்படும்.

OR

- ஒவ்வொரு நிரல்களும் நிரைகளும் இடைவெட்டும் பகுதி ஒரு தனித்த பெறுமதியை (single or atomic value) கொண்டிருத்தல்

## Example 1



## Example 2

Branch_No	BranchAddress	TelNo
B001	Jaffna	021 222 4536 , 021 222 6754
B002	Chavakacheri	021 222 1111
B003	Chankanai	021 222 9876 , 021 222 3434
B004	Nelliyadi	021 222 4536 , 021 222 6750

Branch_No	BranchAddress
B001	Jaffna
B002	Chavakacheri
B003	Chankanai
B004	Nelliyadi

Branch_No	TelNo
B001	021 222 4536
B001	021 222 6754
B002	021 222 1111
B003	021 222 9876
B003	021 222 3434
B004	021 222 4536
B004	021 222 6750

## 2NF [Second Normal Form]

- அட்டவணை ஒன்றில் 2ம் செவ்வன் வடிவில் இருப்பதற்கு பின்வரும் நிபந்தனைகளை பூர்த்தி செய்தல் வேண்டும்.
  - அட்டவணை முதலாம் செவ்வன் வடிவில் இருப்பதுடன் (1NF)
  - அது பகுதியளவான தங்கியிருத்தலை ( partial dependencies) கொண்டிருத்தல் கூடாது.

### Example 1

DoctorID	DoctorName	PatientID	PatientName	Date
D001	Kumar	P002	Smith	10/8/19
D001	Kumar	P001	Ford	5/9/19
D002	Robinsan	P002	Smith	10/8/19

<u>DoctorID</u>	DoctorName
D001	Kumar
D002	Robinsan

<u>PatientID</u>	PatientName
P001	Ford
P002	Smith

<u>DoctorID</u>	<u>PatientID</u>	Date
D001	P002	10/8/19
D001	P001	5/9/19
D002	P002	10/8/19

DoctorID + PatientID = Composite Key

DoctorID → DoctorName

PatientID → PatientName

## Example 2

StaffNo	BranchNo	Branch Address	Name	Position	HoursPer Week
S4555	B002	Chankanai	Vimal	Assistant	19
S4555	B004	Nelliyadi	Vimal	Assistant	9
S4612	B002	Chankanai	Hakeem	Assistant	14
S4612	B004	Nelliyadi	Hakeem	Assistant	10

Branch		Staff		
StaffNo	Branch Address	StaffNo	Name	Position
B002	Chankanai	S4555	Vimal	Assistant
B004	Nelliyadi	S4612	Hakeem	Assistant

Branch_Staff		
StaffNo	BranchNo	HoursPer Week
S4555	B002	19
S4555	B004	9
S4612	B002	14
S4612	B004	10

**BranchNo + StaffNo = Composite Key**

**BranchNo → BranchAddress**

**StaffNo → Name,Position**

### ❖ Insertion Anomaly

- புதிய புத்தகம் ஒன்றின் விபரத்தை பதிவு செய்யும் போது மாணவர் விபரம் இல்லாது அட்டவணையில் உள்நுழைக்க முடியாது.

## 3NF [Third Normal Form]

- அட்டவணை ஒன்று 3NF ல் இருப்பதற்கு பின்வரும் நிபந்தனைகளை திருப்பதிப்படுத்த வேண்டும்.
- அது 2NF ல் இருப்பதுடன் நிலைமாற்ற தங்கியிருத்தலை கொண்டிருத்தல் கூடாது.

OR

- அட்டவணையின் ஒவ்வொரு தீர்மானிக்கும் புலமும் (Determinant) ஓர் சாவிடாக இருத்தல் வேண்டும்.(Candidate key/Primary key)

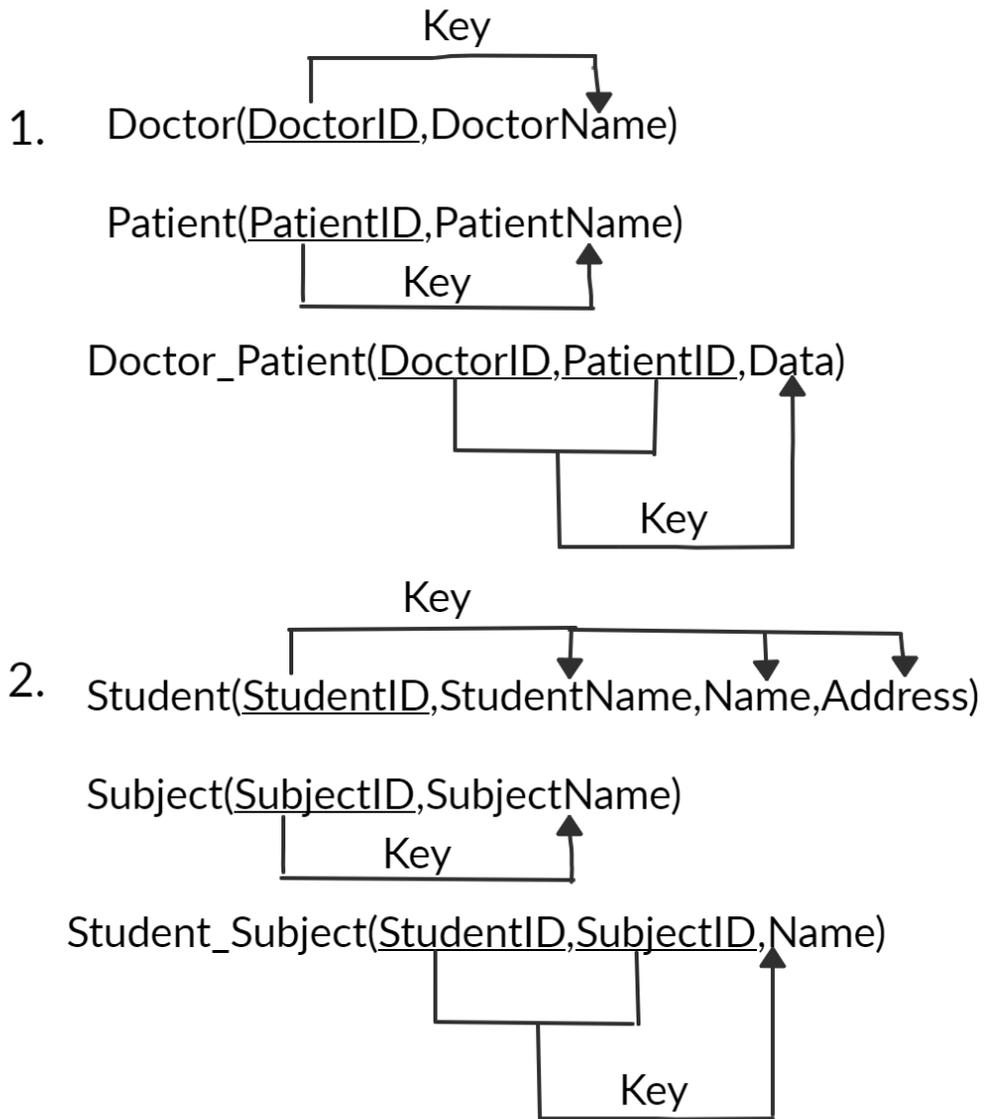
BranchNo	Branch Address	TelNo	Mgr StaffNo	Name
B001	Jaffna	021222 1111	S01	Peter
B002	Chankanai	021222 2222	S02	John
B003	Chavakacheri	021222 3333	S03	Kamal
B004	Nelliyadi	021222 4444	S04	Pieris

Branch

<u>BranchNo</u>	Branch Address	TelNo	Mgr StaffNo
B001	Jaffna	021222 1111	S01
B002	Chankanai	021222 2222	S02
B003	Chavakacheri	021222 3333	S03
B004	Nelliyadi	021222 4444	S04

Branch\_Staff

<u>BranchNo</u>	Name
B001	Peter
B002	John
B003	Kamal
B004	Pieris



## ஒருங்கமைவு கட்டுப்படுத்திகள் (integrity constraints)

### ❖ உள்பொருள் ஒருங்கமைவு (entity integrity)

- முதன்மைசாவி பண்பினது பெறுமதி வெற்று பெறுமதியாக (Null value) இருக்க முடியாது.
- அது தனித்தனியான தொடர்புகளின் மீது குறிக்கப்படுகிறது.

### ❖ மேற்பொருள் ஒருங்கமைவு (referential integrity)

- அட்டவணை ஒன்றின் அந்நியச்சாவி பெறுமதிகள் இன்னொரு அட்டவணையின் முதன்மை சாவியின் பெறுமதிகளை ஒத்ததாக இருத்தல் வேண்டும்.

### ❖ ஆள்கள ஒருங்கமைவு (Domain Integrity)

- அட்டவணை ஒன்றின் ஒவ்வொரு பண்புகளும் அதனோடு இணைந்த ஒரே ஆள்களத்தில் இருத்தல் வேண்டும்

## பலவீனமான உள்பொருள் (Weak entity)

- தொடர்புநிலை தரவுத்தளமொன்றில் உள்பொருள் ஒன்றினை அதன் பண்புகளை மாத்திரம் பயன்படுத்தி தனித்துவமாக அடையாளப்படுத்த முடியாத சந்தர்ப்பம் இதுவாகும்.
- ஆகவே பலவீனமான உள்பொருள் ஒன்றின் முறைமை சாவியினை உருவாக்குவதற்கு அதன் பண்புகளுடன் இன்னொரு உள்பொருளின் அன்னியசாவியினையும் பயன்படுத்தல் வேண்டும் .



## ( EER Diagram Enhanced ER Diagram )

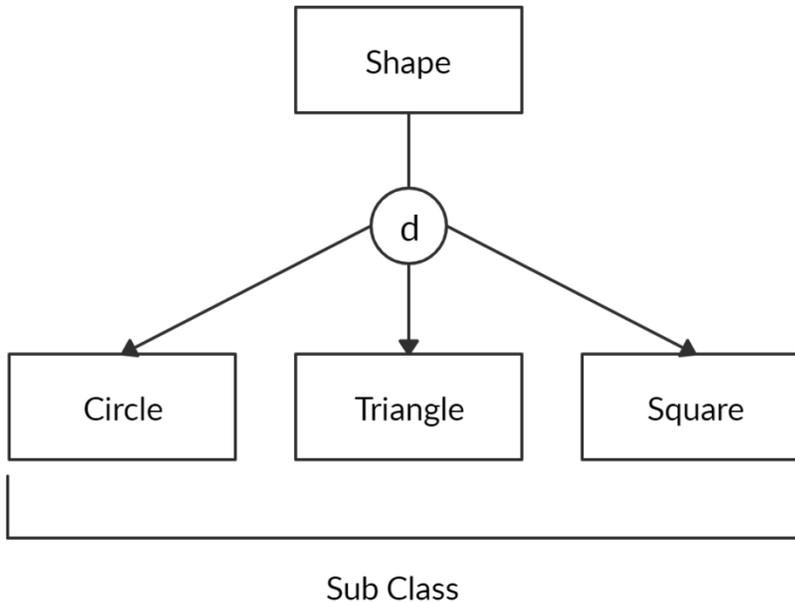
- **EER** என்பது ஓர் உயர்நிலை தரவு மாதிரியம் ஆகும்.
- அது (**ER**) வரிப்படத்தினது நீட்சிப்படுத்தப்பட்ட வசதிகளை கொண்டிருக்கும் பின்வரும் எண்ணக்கருக்களை சாட்சிப்படுத்துவதற்கு இந்த வரிப்பட உத்தியினை பயன்படுத்த முடியும்.

1. உப வகுப்பு/மீவகுப்பு
2. சிறப்பாக்கம்/பொதுமை
3. கூட்டு/வகை
4. ஒருங்கிணைத்தல்

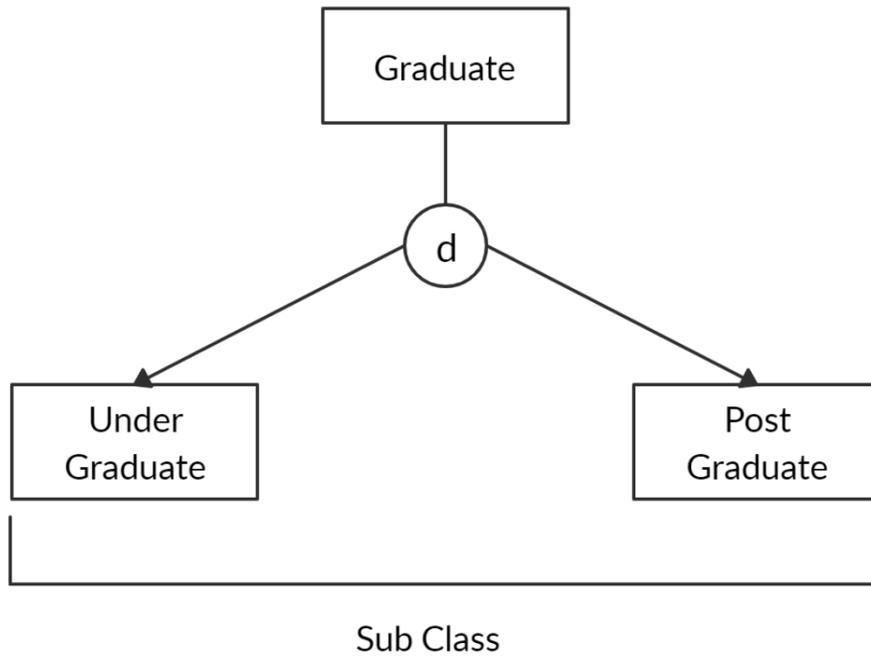
### 1. உப வகுப்பு /மீவகுப்பு (Sub class and super class)

- உப வகுப்பு மற்றும் மீவகுப்பு ஆகியவற்றுக்கிடையிலான தொடர்புடைமையினை (Inheritance) குறிக்கின்றது.
- உப வகுப்பு மற்றும் மீவகுப்பு ஆகியவற்றுக்கிடையிலான தொடர்புடைமை (Inheritance) யினால் குறிக்கப்படும்

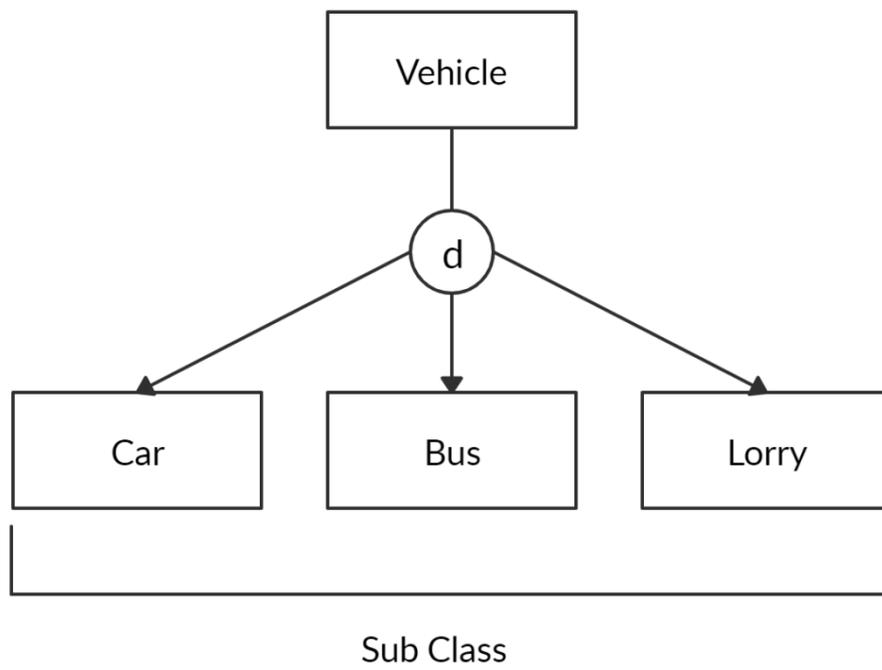
1.



2.



3.



## 2. மீவகுப்பு (super class)

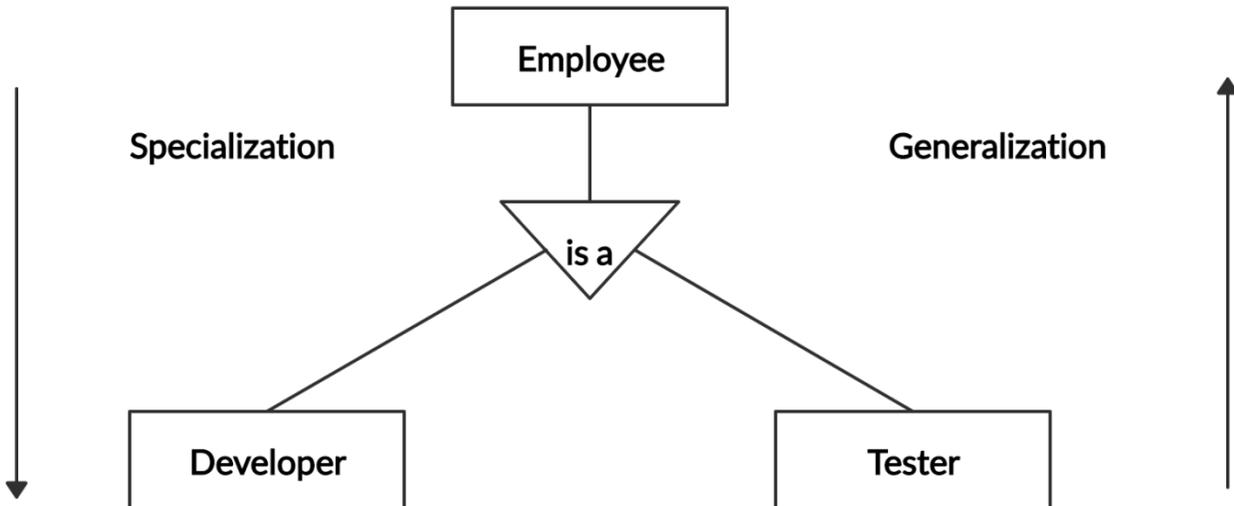
- மீவகுப்பானது ஓர் உள்பொருள் வகையாகும் அது ஒன்று /அதற்கு மேற்பட்ட உப வகுப்புக்களுடன் (**sub classes**) தொடர்புடையமைவினை கொண்டிருக்கும்.

## 3. உப வகுப்பு (sub classes)

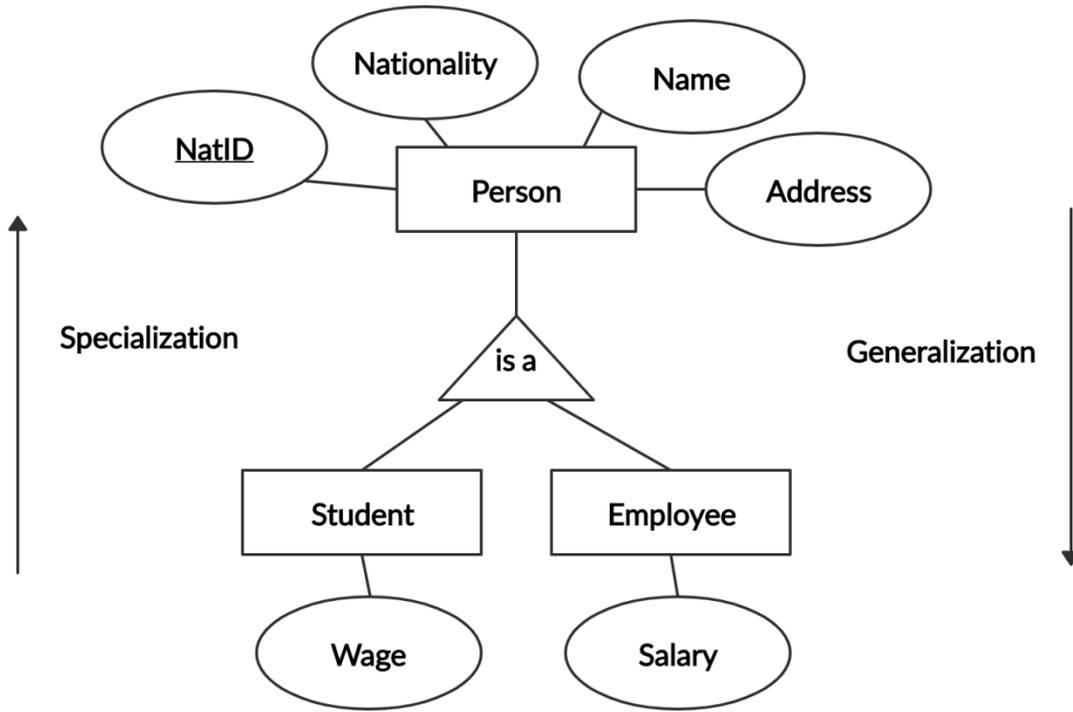
- உப வகுப்பு என்பது ஓர் தனித்துவமான பண்புகளை கொண்ட உள்பொருள்களின் தொகுதி ஆகும் .உப வகுப்பு ஆனது மீ வகுப்பிலிருந்து அதன் பண்புகளை பெறுகின்றது

## ❖ சிறப்பாக்கம் /பொதுமை (Specialization and generalization)

1.



2.



3.

