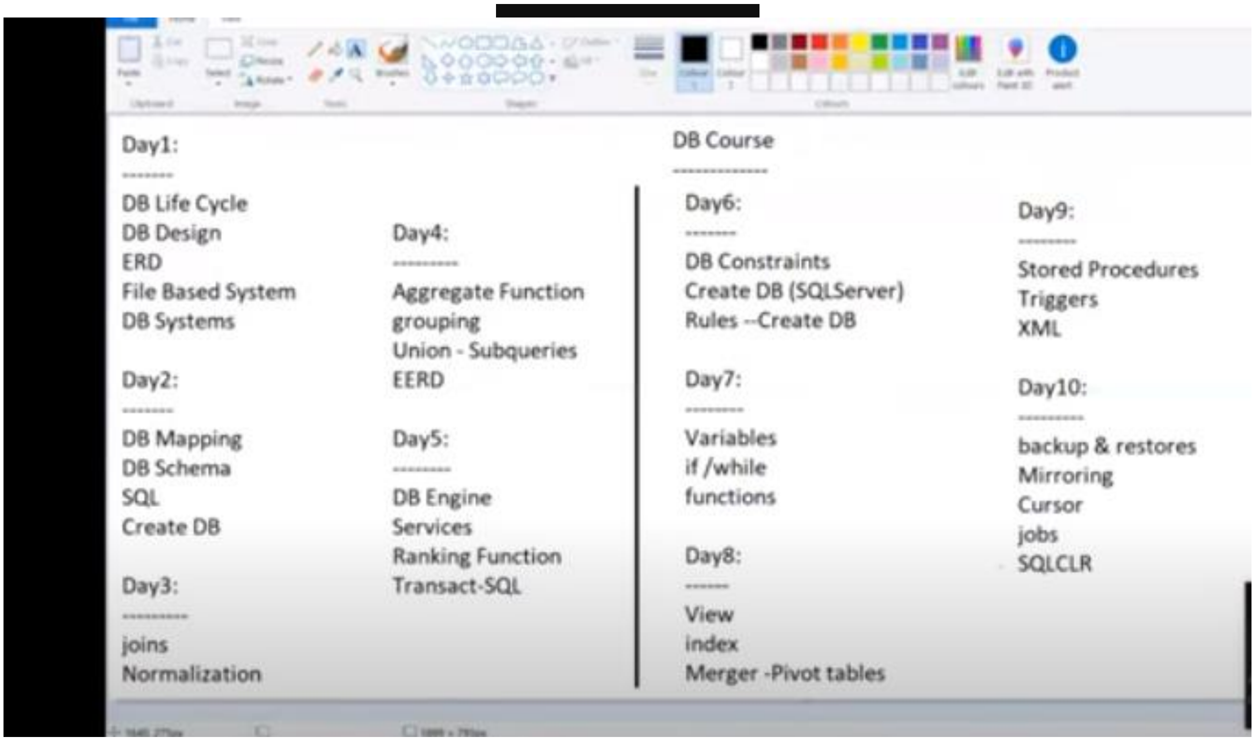
SQL



DB Course

Day1:

DB Life Cycle
DB Design
ERD
File Based System
DB Systems

Day2:

DB Mapping
DB Schema
SQL
Create DB

Day3:

----

joins
Normalization

Day4:

Aggregate Function
grouping
Union - Subqueries
EERD

Day5:

DB Engine
Services
Ranking Function
Transact-SQL

Day6:

DB Constraints
Create DB (SQLServer)
Rules --Create DB

Day7:

Variables
if /while
functions

Day8:

View
index
Merger -Pivot tables

Day9:

Stored Procedures
Triggers
XML

Day10:

----

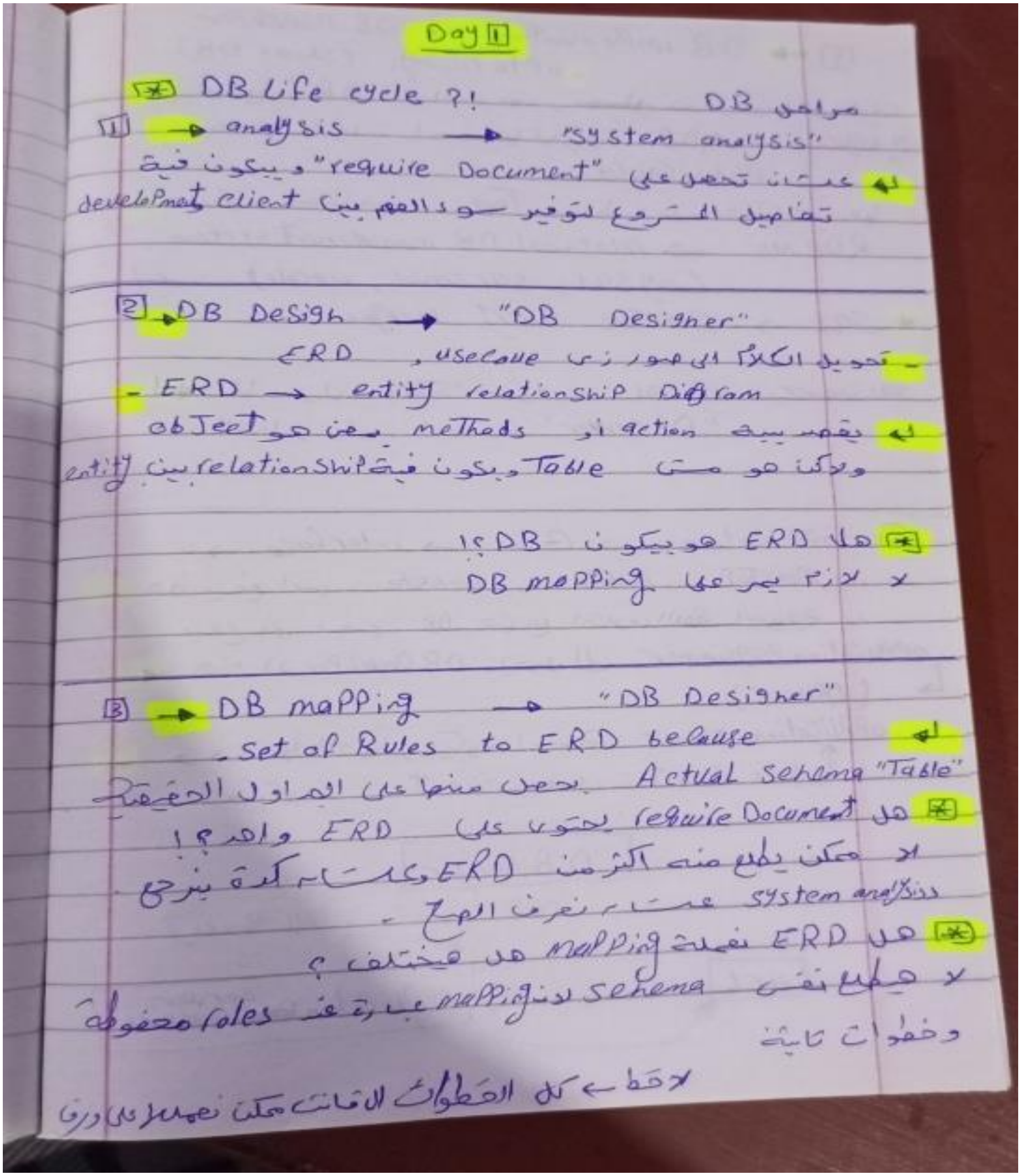
backup & restores
Mirroring
Cursor
jobs
SQLCLR

1-SQL introduction, DB Design, ERD, File Based System, DB system

1)هل كل applications لازم يكون ليها DB ؟!

لا زي office , calculator , static website(html,css,js)

ولاكن 95% من Big applications معتمدة علي DB



Day [1]

[*] DB Life cycle ?! مراحل DB

[1] → analysis → "system analysis"

↲ عشان تحصل على "require Document" ويكون فية
تفاصيل المشروع لتوفير سوء الفهم بين client development

[2] → DB Design → "DB Designer"

- تحويل الكلام الى صور زى usecase , ERD
- ERD → entity relationship Diagram

↲ يقصد بية action او methods يعنى هو object
ولكن هو مش Table ويكون فية relationship بين entity

[*] هلا ERD هو بيكون DB ؟!
x لازم يمر عليها DB mapping

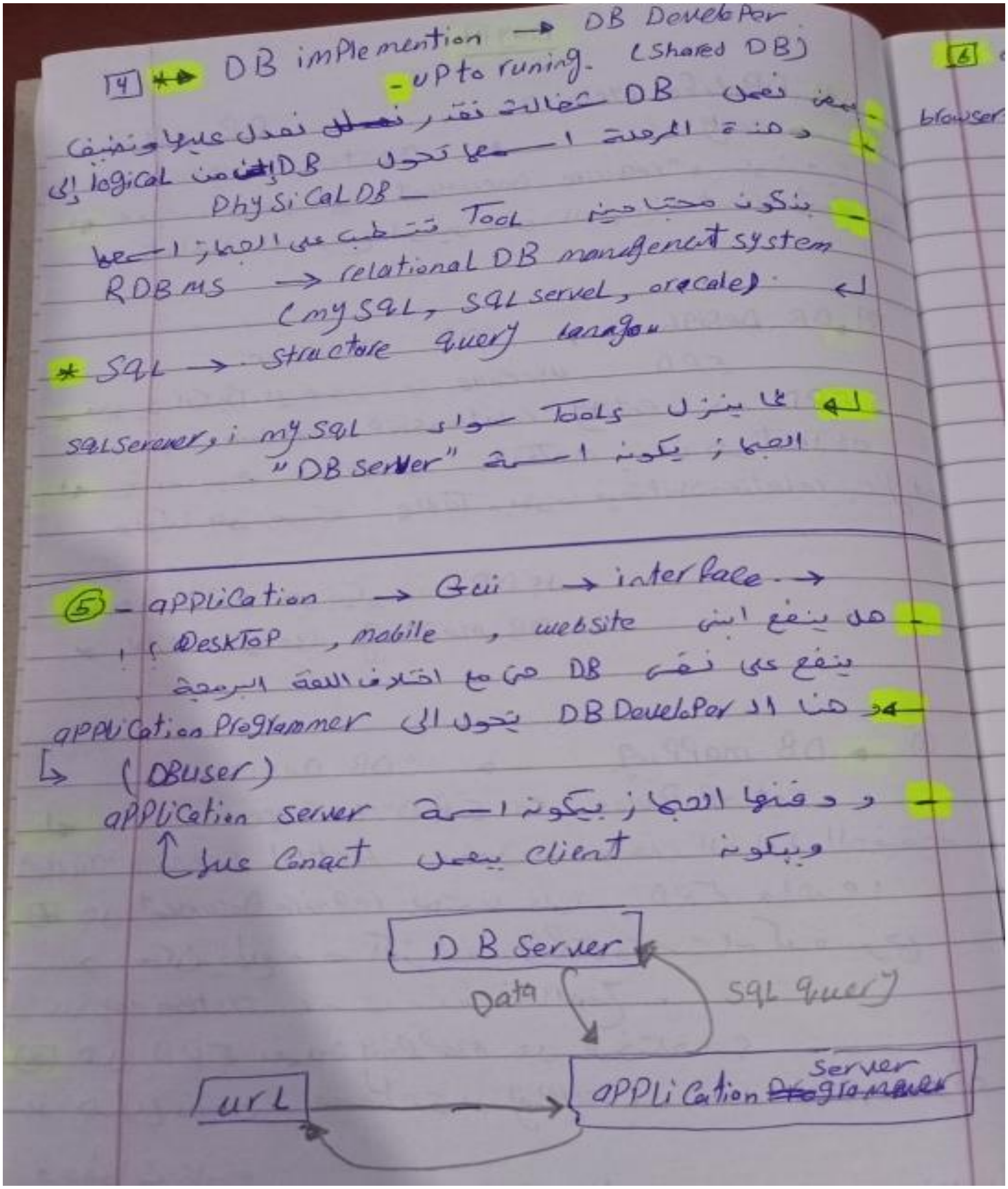
[3] → DB mapping → "DB Designer"

↲ set of Rules to ERD because
Actual schema "Table" بحصل منها على الجداول الحقيقية

[*] هل require Document يحتوى على ERD واحد ؟!
x ممكن يطلع منه اكثر من ERD وعشان كدة بنرجع system analysis عشان نعرف الصح .

[*] هل ERD نفسة mapping هل هيختلف ؟
x هيطلع نفس schema لان mapping عبارة عن roles محفوظة وخطوات ثابتة

x فقط ← كل الخطوات الى فاتت ممكن نعملها على ورق



4 ** DB implemention → DB Developer.
- up to runing. (Shared DB)

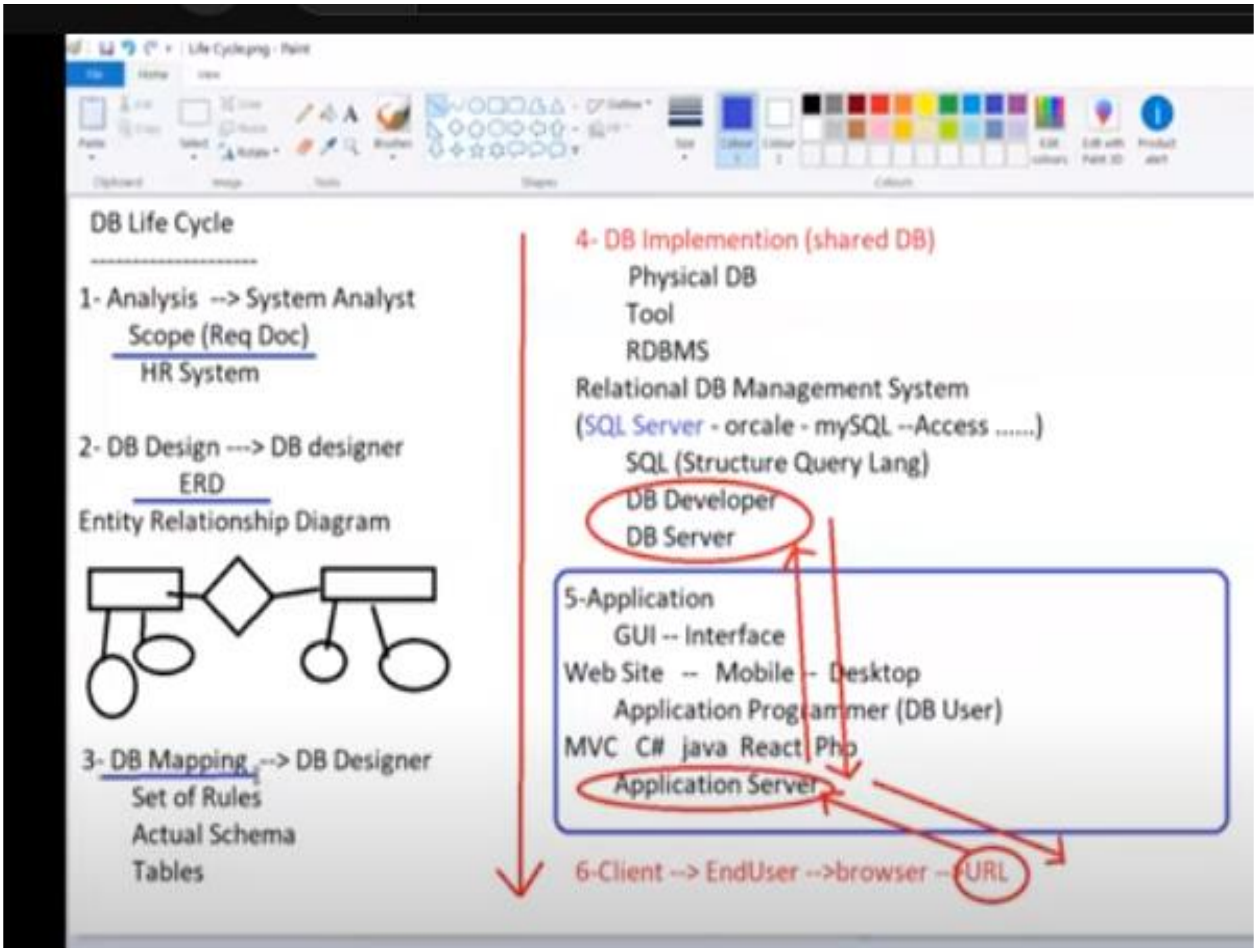
- لمن نعمل DB شغالة نقدر ~~نحلل~~ نعدل عليها ونضيف
- وهذه المرحلة اسمها تحول DB من logical إلى
Physical DB —
- بنكون محتاجين Tool تتسطب على الجهاز اسمها
relational DB mangement system
RDBMS → (mySQL, SQL servel, oracale) ↲

* SQL → structure query langou

- لما ينزل Tools سواء mySQL ، sqlserever
الجهاز بيكون اسمه "DB server"

⑤ - application → Gui → interface →
- هل ينفع ابني website , mobile , Desktop ؟!
ينفع على نفس DB هي مع اختلاف اللغة البرمجية
- هنا الـ DB Developer يتحول الى application Programmer
↳ (DBuser)
- ودقتها الجهاز بيكون اسمه application server
وبيكون client بيعمل Conact ↑ lue

DB Server
Data
SQL query
url
application ~~Programmer~~ Server



DB Life Cycle

1- Analysis --> System Analyst

Scope (Req Doc)

HR System

2- DB Design ---> DB designer

ERD

Entity Relationship Diagram

3- DB Mapping --> DB Designer

Set of Rules

Actual Schema

Tables

4- DB Implemention (shared DB)

Physical DB

Tool

RDBMS

Relational DB Management System

(SQL Server - orcale - mySQL --Access ......)

SQL (Structure Query Lang)

DB Developer

DB Server

5-Application

GUI -- Interface

Web Site -- Mobile -- Desktop

Application Programmer (DB User)

MVC C# java React Php

Application Server

6-Client --> EndUser -->browser --URL

6 client → enduser → browser → url

- هل client ليه علاقة بـ DB ؟! لا ليه علاقة بـ browser
يكتب الـ url

← نعرّ" لأن (orcale او SQLServer) كبيرة و غالية إتجهوا
الى File Based system.

* وهو عبارة عن Text file. الميزة هي رخيصة جدا
ومش عاوز حد خبرة في المجال.

- ولازم الـ file يكون ليه Format

☒ Delimited file

- بيفصل بينهم بـ (,)
"Delimiter"

1, Nora, 25

2, Eman, 20

وهنا انا بفصل من كل واحد بـ طرد
(,) Delimiter

☒ Fixed widTh file

- بيشتغل بعدد Bits معينة

1 Java 20 5D

يعني بعمل read , write
بعدد Bits معينة
يعني id ← 1 ليه 2Bite
Jave ← ليها 5Bite
و 20 ← ليها 3Bite
وهكذا

ايه المشاكل ال هتكون موجودة ؟

1 different search	صعوبة في البحث
2 low Performance	الكفاءة ضعيفة
3 No relationShip	مفيش علاقات .
4 SeParated CoPies.	نسخ منفصلة .
5 No DB integrity	مفيش ترابط بين DB

⑥ DB DuPLiCation

تكرار DB

⑦ long DeuelolPment time.

لو عايزة اتفادى خطأ' ال DuPLiCation لازم اكتب كود
كبيرة علشان اتفادى الخطأ.

⑧ Security. & Permissions. مش مأمنة

⑨ Constrains & rules. مفيش قيود على "Tables"

⑩ No Data quality

يعني مش هيعرف يفرق بين "DataType"

⑪ manual Backup & restore.

⑫ No Standard. مفيش نظام معين ماشيين عليه.

⑬ Different integrate. صعب ~~اخلي~~ اجمع بينهم "2 fileText"
لاختلاف الطريقة.

* ونظرًا لوجود المشاكل الكثيرة ظهر مصطلح
→ "DB System"

~~وهي تحويل TexT file إلى Tables بينها relationships.~~

- هي تحويل Textfile الى Tables بينها relationships

- مميزات نظام "DB System"

① one standard. ليها نفس الشكل "Tables" -

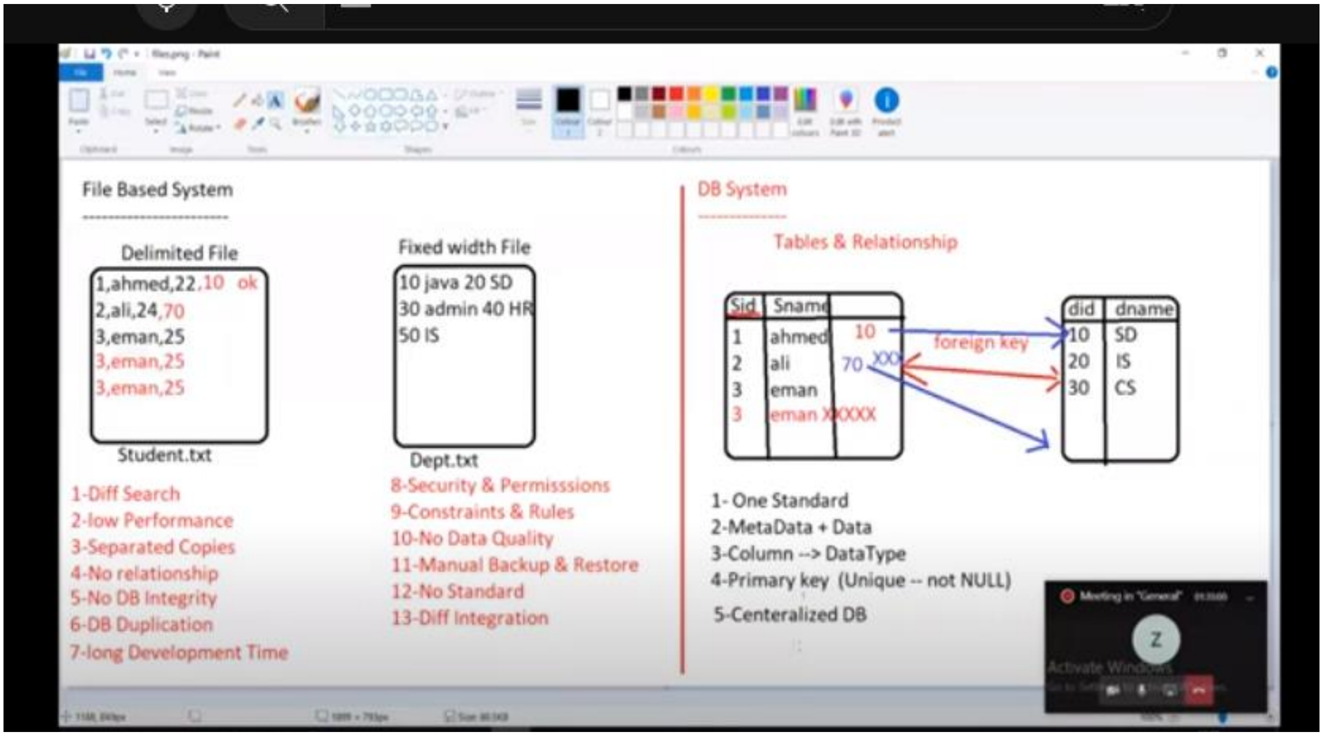
② meta Data + Data الاتنين موجودين جوة DB -

- مقارنة بين Data , metaData ؟!

- metaData → atribute, اسماء Tables, علاقات.

- Data → values "cells" الموجودة جوة

- ماهو الموجود في er diagram , metaData ولا Data ؟
"meta Data"



File Based System

Delimited File

1,ahmed,22,10 ok
2,ali,24,70
3,eman,25
3,eman,25
3,eman,25

Student.txt

Fixed width File

10 java 20 SD
30 admin 40 HR
50 IS

Dept.txt

1-Diff Search
2-low Performance
3-Separated Copies
4-No relationship
5-No DB Integrity
6-DB Duplication
7-long Development Time

8-Security & Permisssions
9-Constraints & Rules
10-No Data Quality
11-Manual Backup & Restore
12-No Standard
13-Diff Integration

DB System

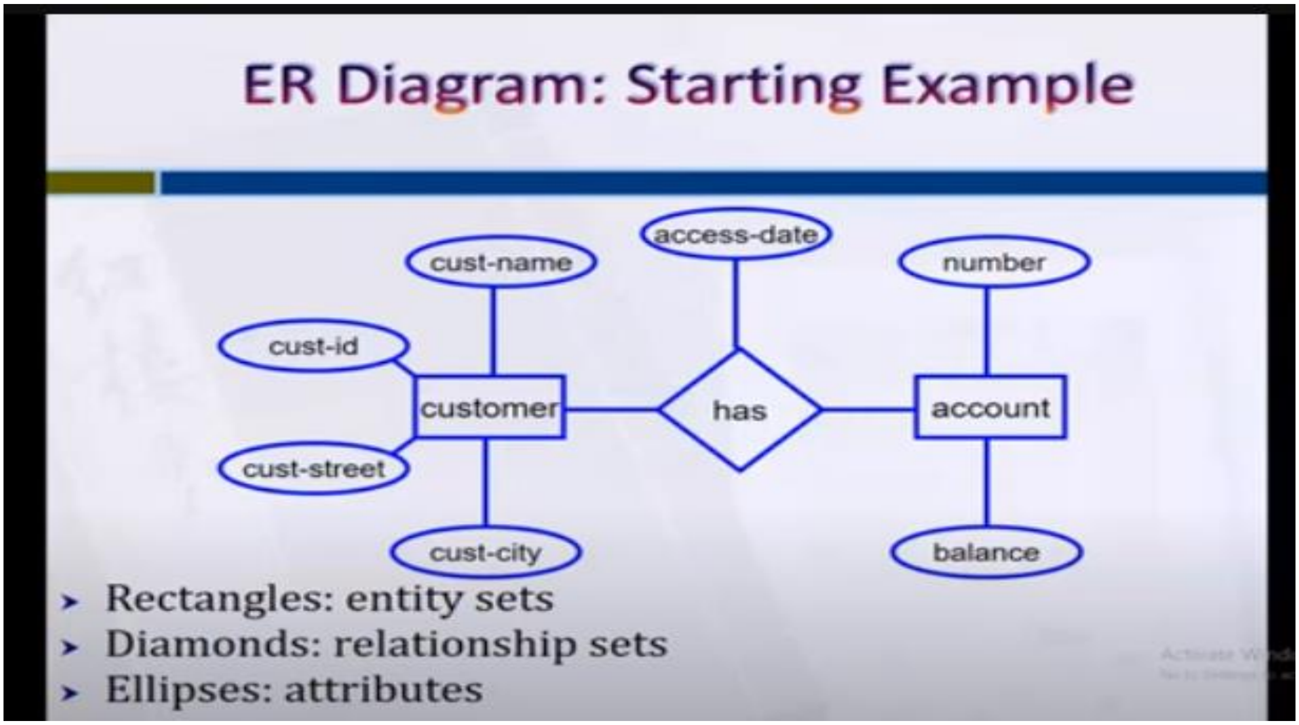
Tables & Relationship

Sid	Sname	
1	ahmed	10
2	ali	70 XXX
3	eman	
3	eman XXXXX	

foreign key

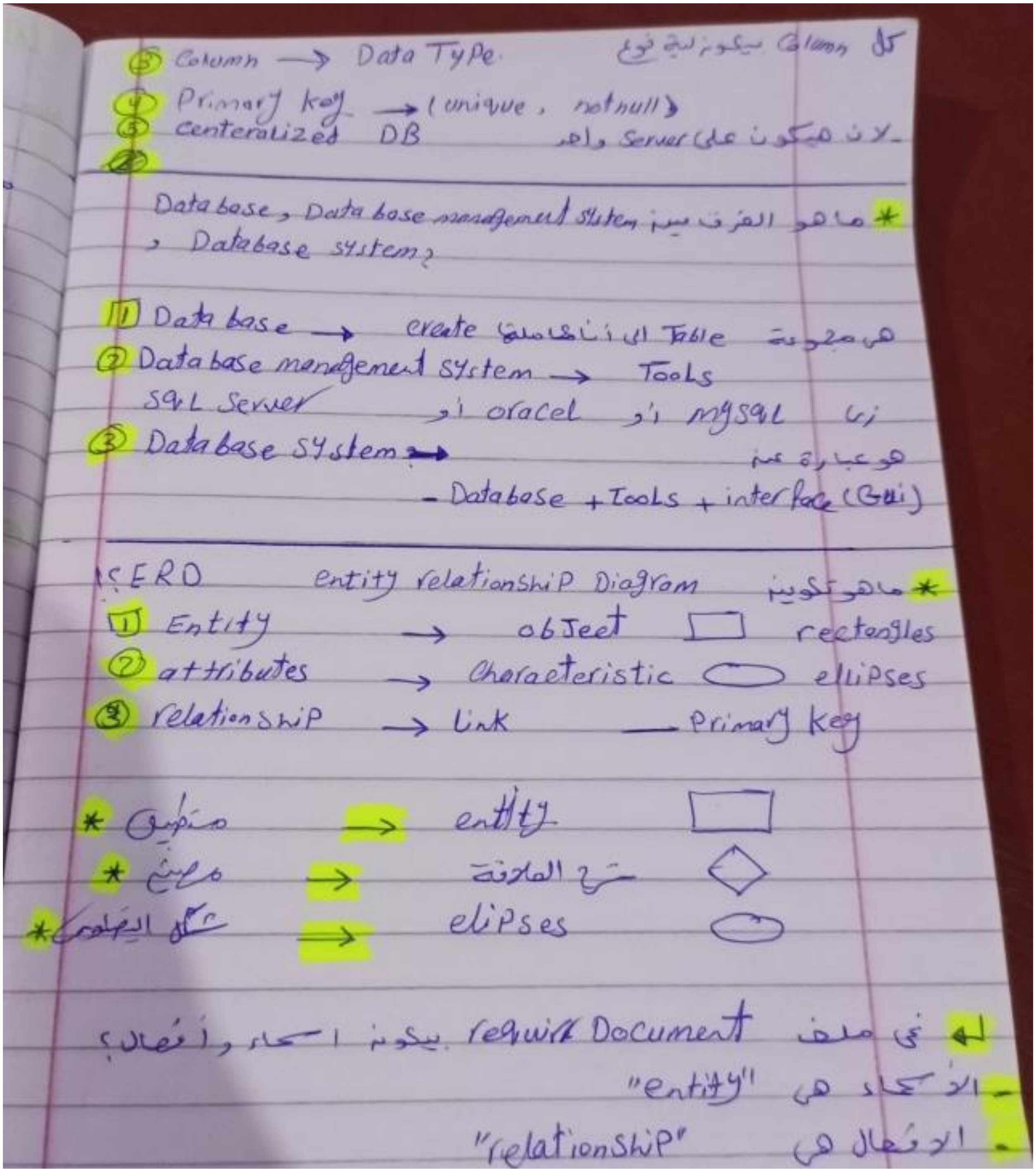
did	dname
10	SD
20	IS
30	CS

1- One Standard
2-MetaData + Data
3-Column --> DataType
4-Primary key (Unique -- not NULL)
5-Centeralized DB



ER Diagram: Starting Example

- Rectangles: entity sets
- Diamonds: relationship sets
- Ellipses: attributes



⑤ Column → Data Type. كل Column بيكون ليه نوع

④ Primary key → (unique , notnull)

③ centeralized DB - لان هيكون على Server واحد

②

---

* ماهو الفرق بين Database, Data base managment system , Database system?

[1] Data base → هي مجموعة Table الى انا هعملها create

② Database manegement system → Tools

زي mysql أو oracel أو SQL Server

③ Database System → هو عبارة عن

- Database + Tools + interface (GUI)

---

* ماهو تكوين ERD: entity relationship Diagram

[1] Entity → object □ rectangles

② attributes → characteristic ⬭ ellipses

③ relationship → link —— Primary key

* مستطيل → entity □

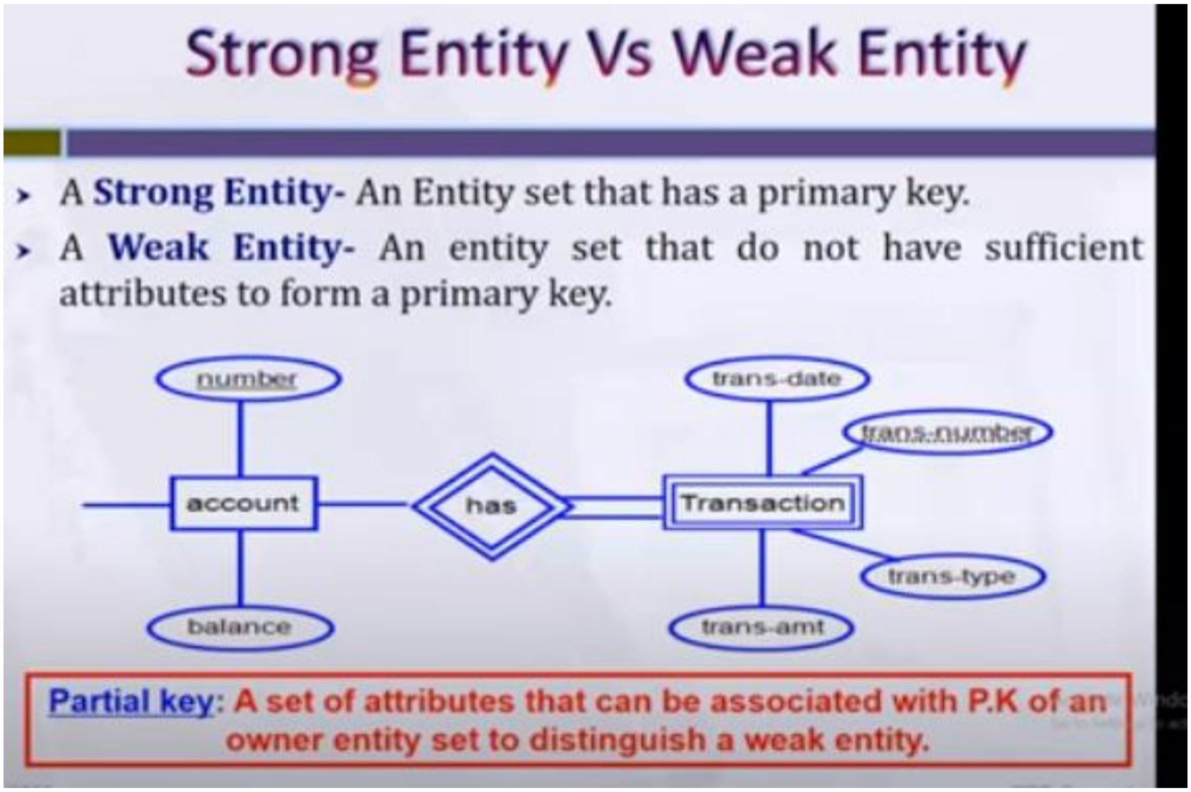
* معين → شرح العلاقة ◇

* شكل البيضاوي → elipses ⬭

لو في ملف require Document بيكون اسماء و أفعال؟

- الأسماء هي "entity"

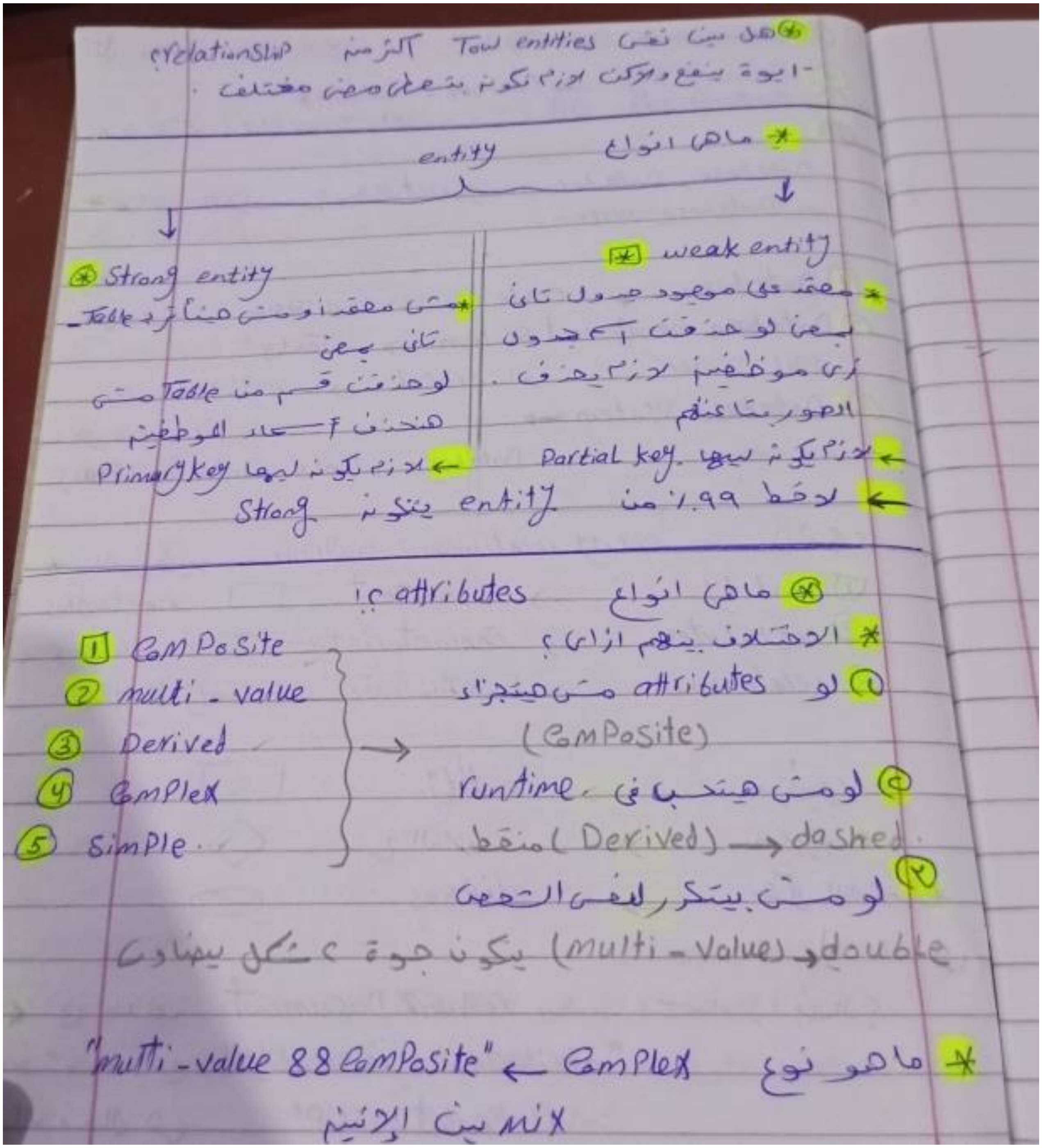
- الأفعال هي "relationship"



Strong Entity Vs Weak Entity

- A **Strong Entity-** An Entity set that has a primary key.
- A **Weak Entity-** An entity set that do not have sufficient attributes to form a primary key.

Partial key: A set of attributes that can be associated with P.K of an owner entity set to distinguish a weak entity.



* هل ينفع نعمل Tow entities اكتر من relationship؟
- ايوة ينفع ولكن لازم تكون بشكل معين مختلف .

* ماهي انواع entity

☒ weak entity

* معتمد على موجود جدول تاني يعني لو حذفت ال جدول زي موظفين لازم يحذف الصور بتاعتهم .

← لازم يكون بيها Partial key.

* Strong entity

* مش معتمد ومش هيأثر ف Table تاني يعني

لو حذفت قسم من Table مش هنحذف أعداد الموظفين

← لازم يكون ليها Primary key

← لاحظ ٩٩٪ من entity بيكون Strong

* ماهي انواع attributes؟!

1. Composite
2. multi - value
3. Derived
4. Complex
5. Simple.

→

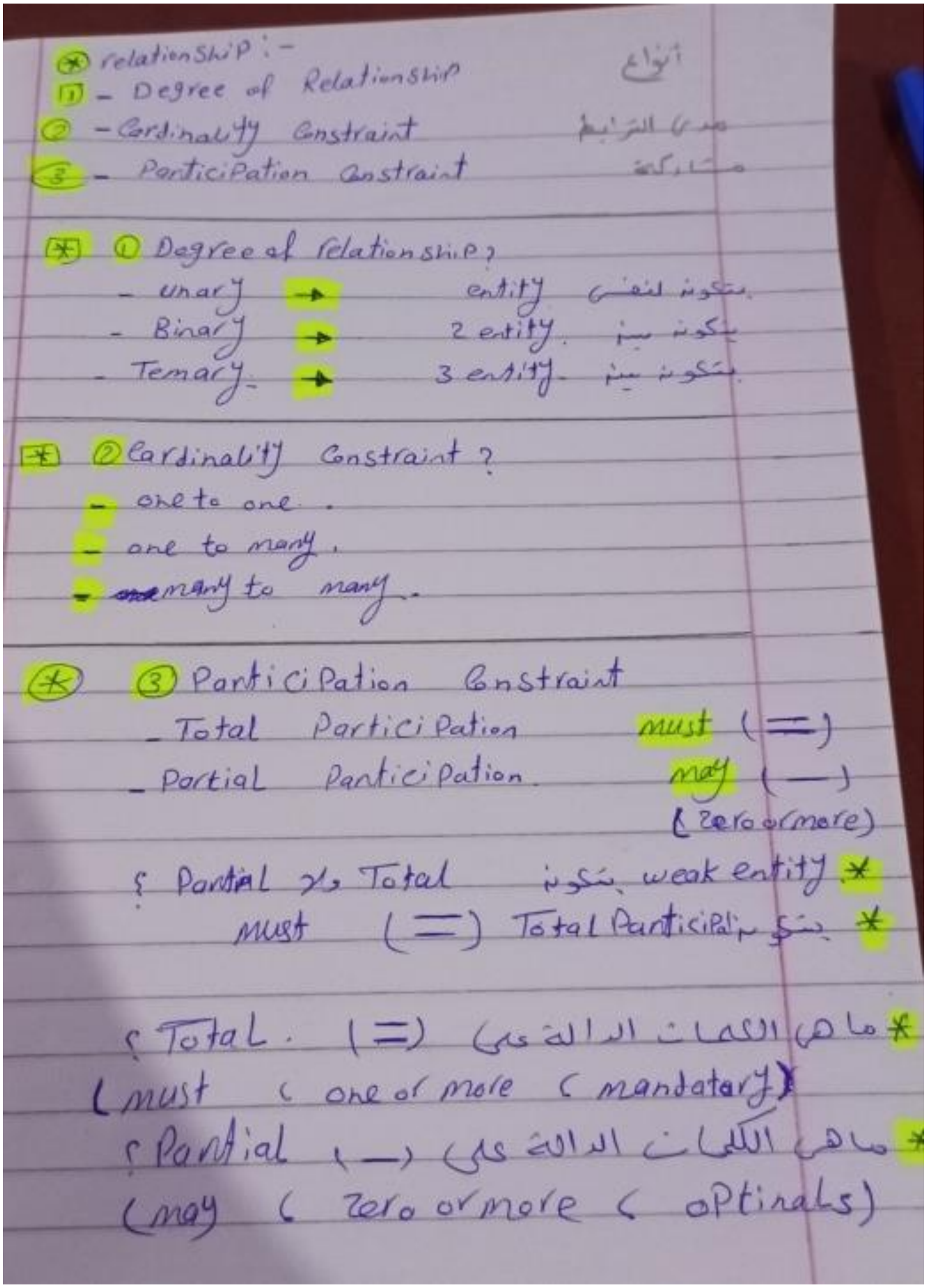
* الاختلاف بينهم ازاي؟

① لو attributes مش هيتجزاء (Composite)

② لو مش هيتحسب في runtime. (Derived) منقط → dashed.

③ لو مش بيتكرر لنفس الشخص (multi - Values) يكون جوة ف شكل بيضاوي → double

* ماهو نوع Complex ← "multi-value && Composite" mix بين الاتنين



(*) relationship :- أنواع

1 - Degree of Relationship

2 - Cardinality Constraint مدى الترابط

3 - Participation Constraint مشاركة

(*) ① Degree of relationship ?

- unary → entity بتكون لنفسي

- Binary → 2 entity. بتكون بين

- Temary. → 3 entity. بتكون بين

(*) ② Cardinality Constraint ?

- one to one .

- one to many .

- ~~one~~ many to many .

(*) ③ Participation Constraint

- Total Participation must (=)

- Partial Panticipation. may (—)

(zero or more)

* weak entity بتكون Total ولا Partial ؟

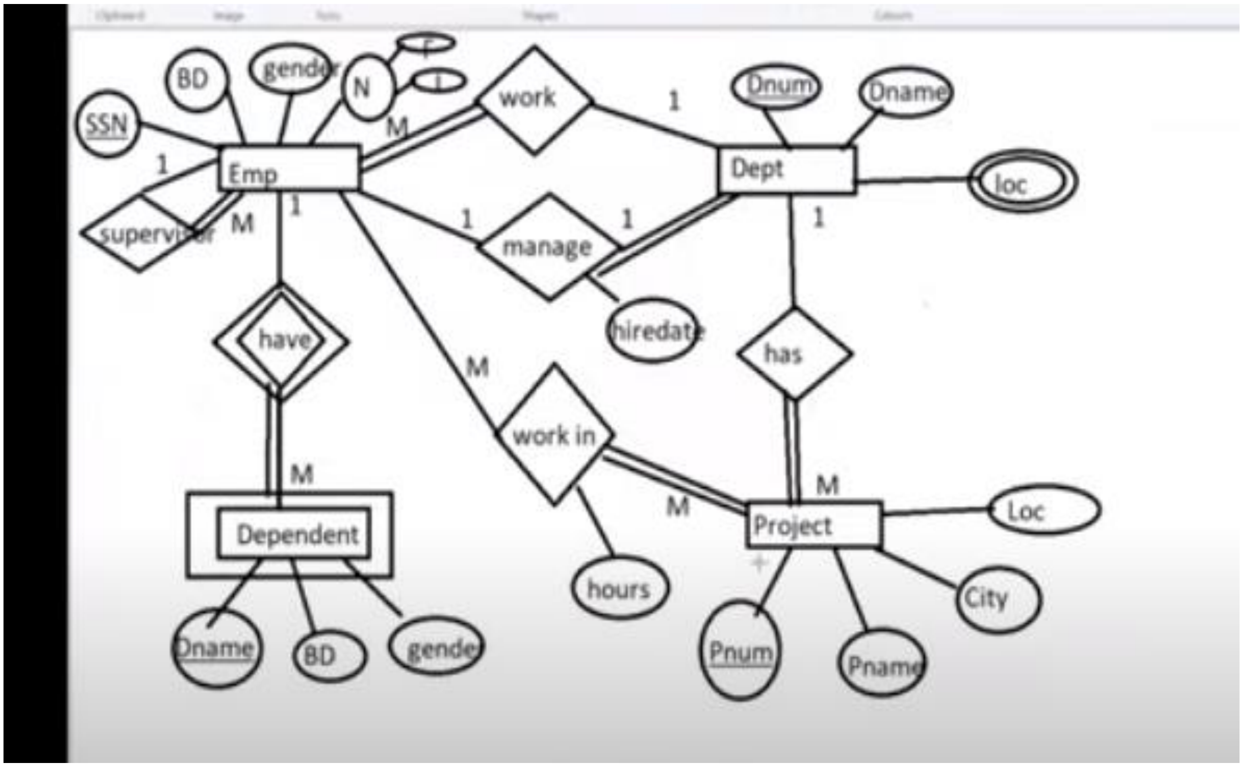
* بشكل Total Participation (=) must

* ماهي الكلمات الدالة على (=) Total ؟

(must ، one or more ، mandatary)

* ماهي الكلمات الدالة على (—) Partial ؟

(may ، zero or more ، optinals)

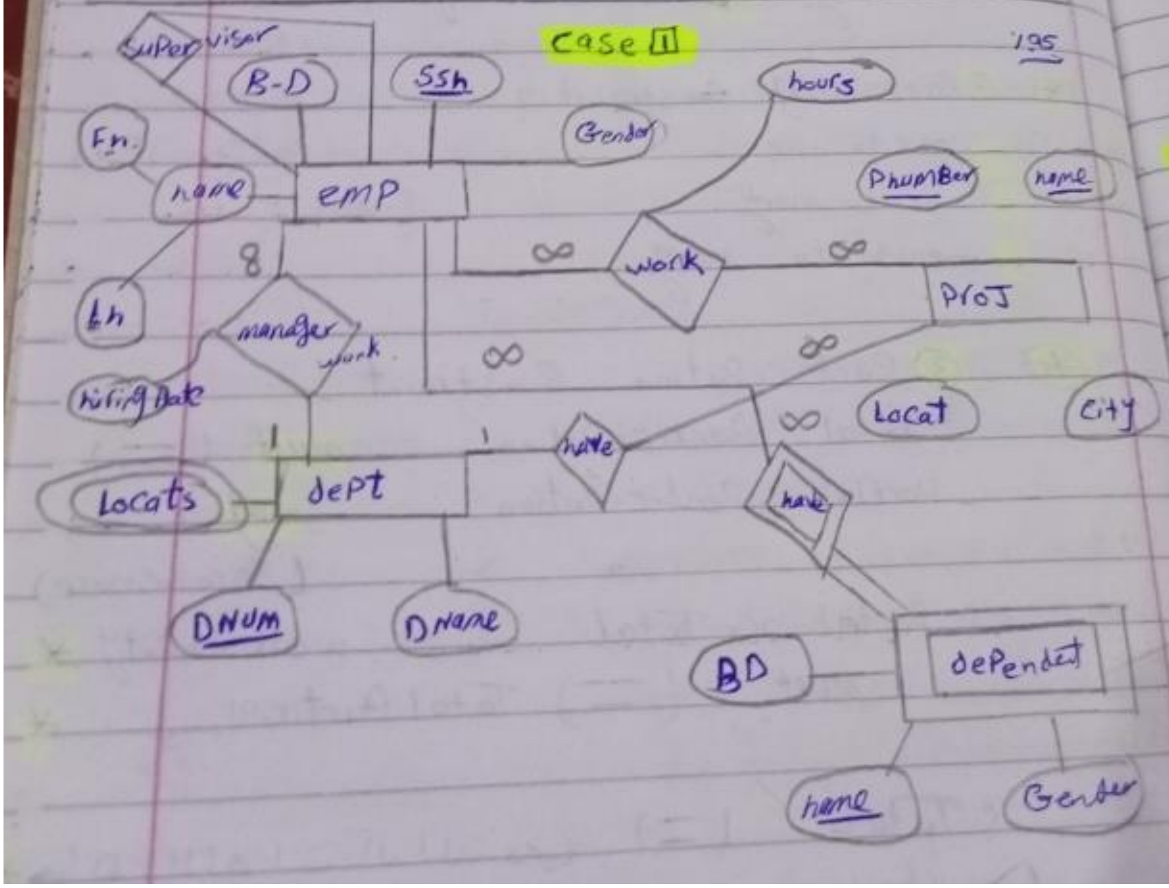


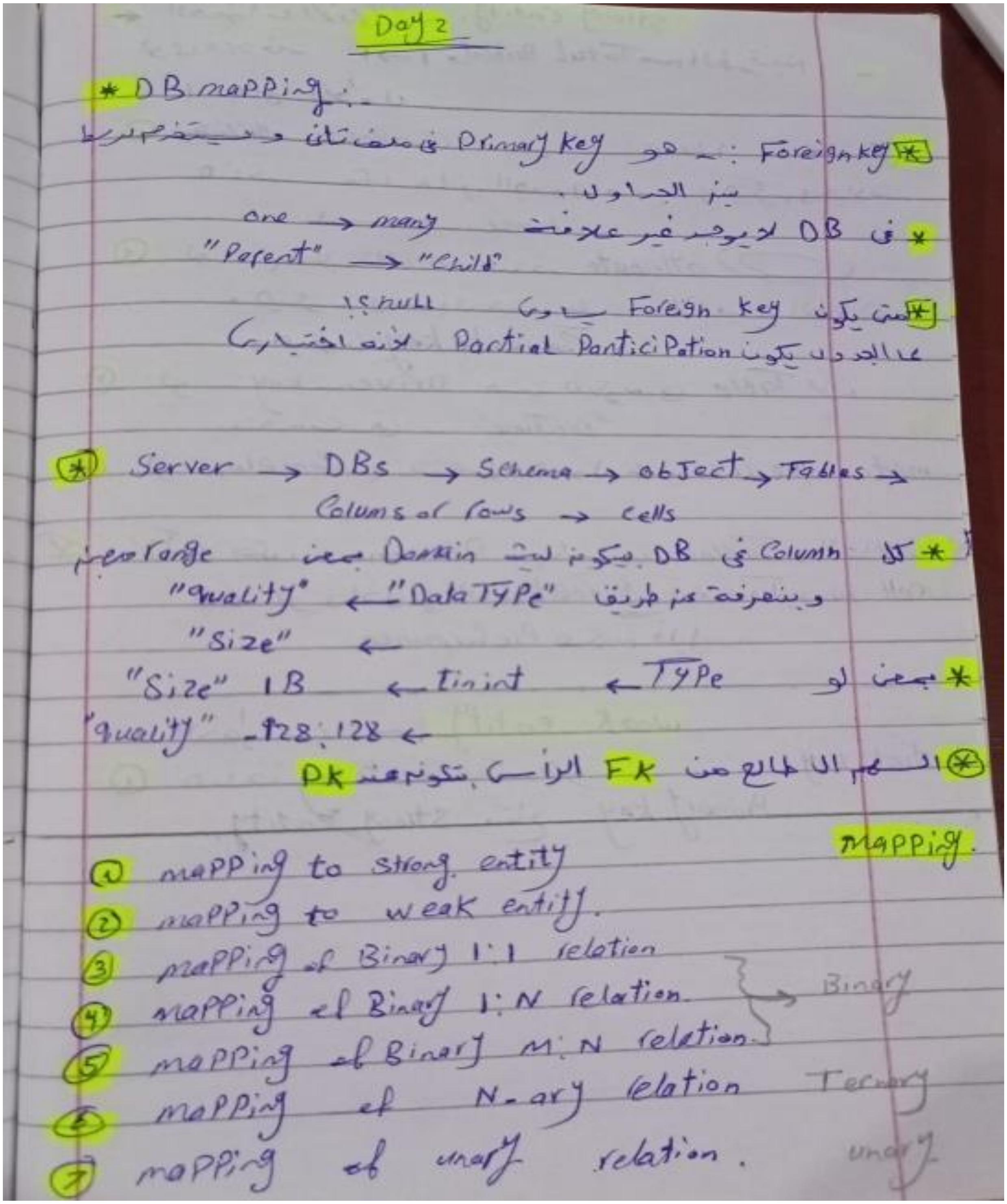
2- SQL Mapping, DB Schema, SQL statements, Creating DB

* أنواع key في DB :!

1. Candidate key → موجودة جوه PK
2. Primary key → under lined.
3. Foreign key
4. ComPosite key ← لو معيش iD بركب atribute مع بعض
5. Partial key
6. Alternate key
7. suPer key

Case 1





Day 2

* DB mapping :-

* Foreign key :- هو Primary key في جدول تاني و بيستخدم للربط بين الجداول.

* في DB لا يوجد غير علاقة one → many
"Parent" → "Child"

* متى يكون Foreign key بيساوي null ؟!
عالجدول يكون Partial Participation لأنه اختياري

* Server → DBs → Schema → object → Tables →
Colums or rows → cells

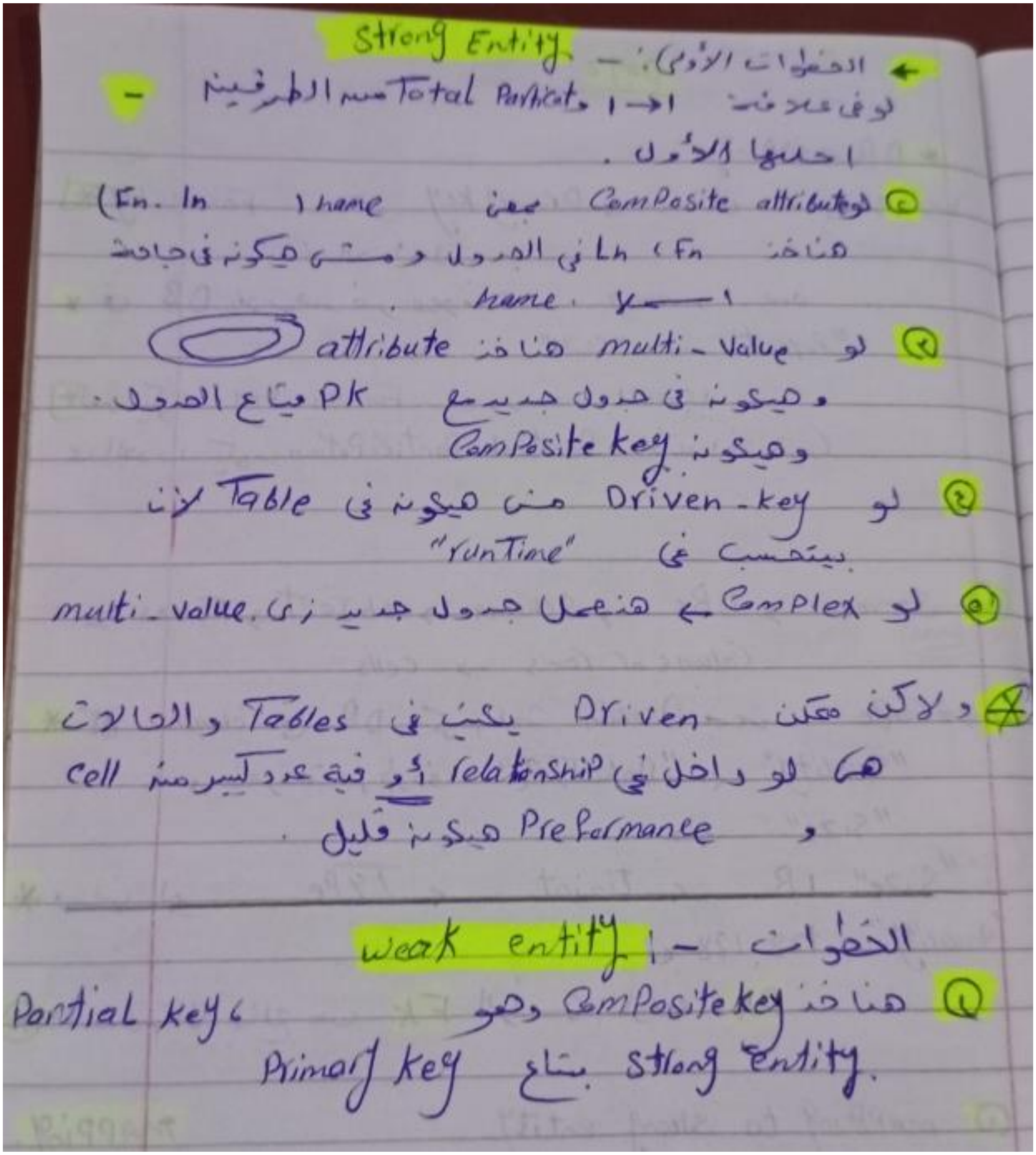
* كل Column في DB بيكون ليه Domain بمعنى range معين
وبنعرفه عن طريق "DataType" → "quality"
→ "Size"

* بمعنى لو Type → tinint → 1B "Size"
→ -128 : 128 "quality"

* السهم الطالع من FK الرأس بتكون عند PK

mapping.

1. mapping to strong entity
2. mapping to weak entity.
3. mapping of Binary 1:1 relation
4. mapping of Binary 1:N relation → Binary
5. mapping of Binary M:N relation.
6. mapping of N-ary relation Ternry
7. mapping of unary relation. unary



← الخطوات الأولى :- **Strong Entity**

لو في علاقة 1→1 Total Particat من الطرفين -
احطها الأول .

① لو Composite attribute يعني (Fn. ln ، name
هناخد Fn ، Ln في الجدول ومش هيكون في حاجة
اسمها name .

② لو multi-Value هناخد attribute ◎
هيكون في جدول جديد مع PK بتاع الجدول .
وهيكون Composite key

③ لو Driven-key مش هيكون في Table لأنه
بيتحسب في "runTime"

④ لو Complex ← هنعمل جدول جديد زي multi-value.

✱ ولكن ممكن Driven يحط في Tables والحالات
هي لو داخل في relationship أو فيه عدد كبير من cell
، هيكون Preformance قليل .

الخطوات :- **weak entity**

① هناخد Composite key وهو Partial key ،
Primary key بتاع Strong Entity.

⊛ Binary oneto one → "٦ حالات"

① onEto one and 2 mandatory (Total)
له هيتحول إلى جدول واحد يعني 2entity هيكونوا في
جدول واحد

② oneto one x optional , y mandatory
له هينفصلة في جدولين ولكن FK هيكون في y mandatory

③ oneto one x , y optional
له هيكون ٣جداول و الجدول الثالث يحتوى على
PKx ، PKy

④ one to many x optional, y mandatory
له هيكون جدولين و FK في many ~~mandatory~~
حتى لو مش mandatory

⑤ one to many 2 optional
له هيكون ٣جداول والجدول ٣ هيكون فيه PKx ، PKy
~~⑥~~ و primary هيكون ال من نحية many

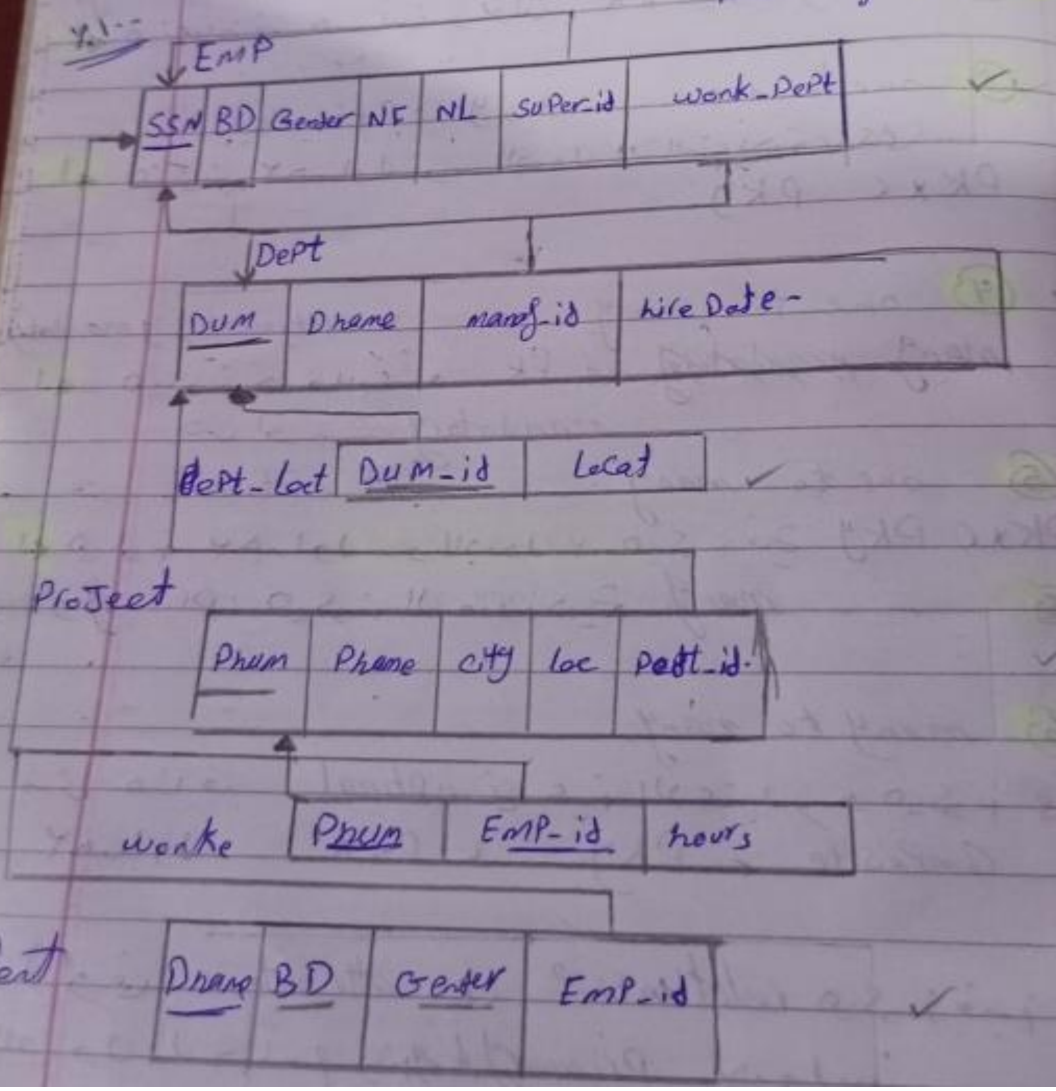
⑥ many to many.
مش هناخد optional في عين الاعتبار، و هيكون في
٣جداول PKx ، PKy ← Composite.

* لاحظ لو في attribute على relationship هيكون فين؟!
في الجدول الجامع ينفع primary key لجدولين

له هيتحول الى جداول على طول ومش هناخد في عين الاعتبا، ~~II~~ ~~Iran~~ Ternary $n>2$ "Cardinality ، Partical"

له FK بيكون في نفس الجدول. * unary relation ship

Case II "mapping"



%100

EMP

SSN	BD	Gender	NF	NL	Super_id	work_Dept

✓

Dept

DUM	Dname	manf_id	hire Date-

✓

Dept_lot

DUM_id	Locat

✓

ProJect

Pnum	Pname	city	loc	Dept_id.

✓

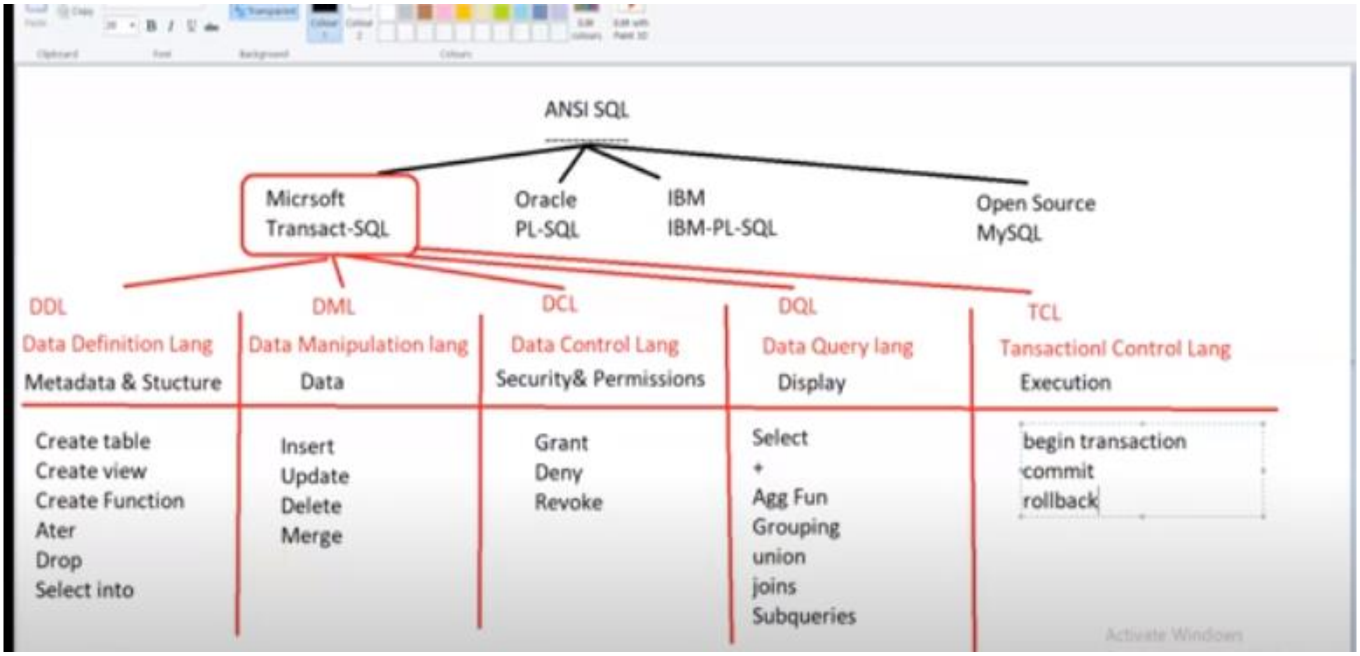
worke

Pnum	EMP_id	hours

de Pent

Dname	BD	Gender	EMP_id

✓



ANSI SQL

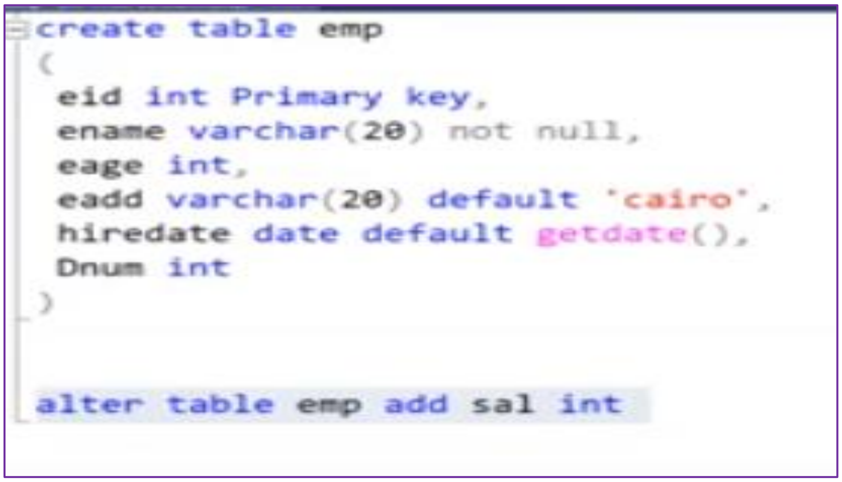
Micrsoft Transact-SQL

Oracle PL-SQL

IBM IBM-PL-SQL

Open Source MySQL

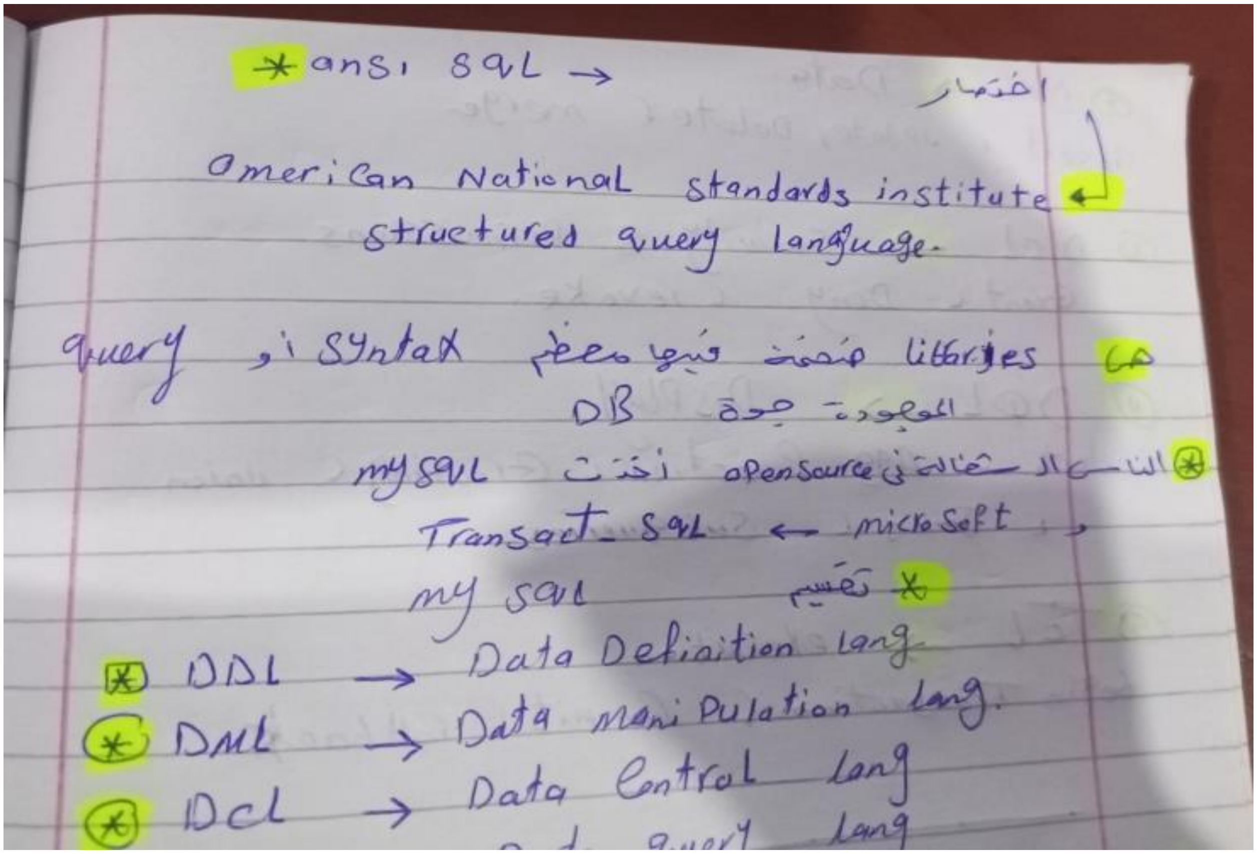
DDL	DML	DCL	DQL	TCL
Data Definition Lang	Data Manipulation lang	Data Control Lang	Data Query lang	Tansactionl Control Lang
Metadata & Stucture	Data	Security& Permissions	Display	Execution
Create table	Insert	Grant	Select	begin transaction
Create view	Update	Deny	*	commit
Create Function	Delete	Revoke	Agg Fun	rollback
Ater	Merge		Grouping	
Drop			union	
Select into			joins	
			Subqueries	



```
create table emp
(
 eid int Primary key,
 ename varchar(20) not null,
 eage int,
 eadd varchar(20) default 'cairo',
 hiredate date default getdate(),
 Dnum int
)


alter table emp add sal int
```

Create table

Alter table to add new attribute



* ansi sql → اختصار

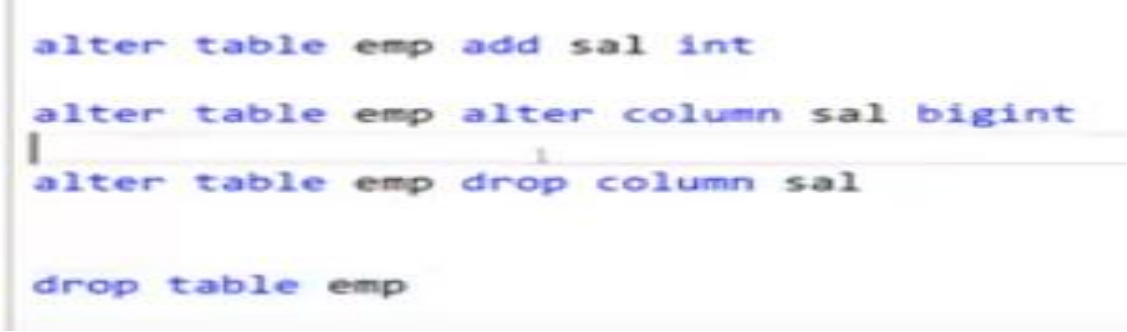
american National standards institute ←
structured query language.

هي libraries منفصة فيها معظم syntax الـ query
الموجودة جوة DB

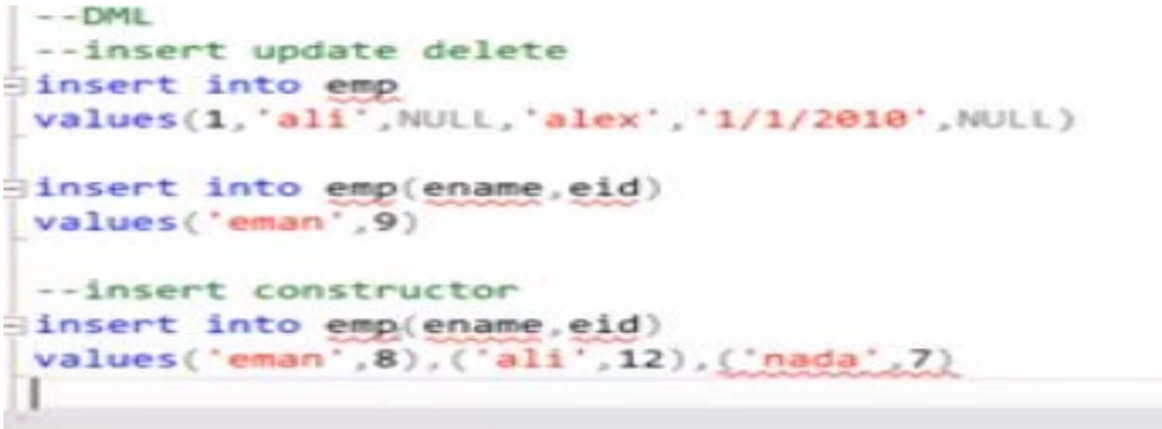
* الناس ال شغالة في opensource أخدت mysql
و microsoft ← Transact_sql

* تقسيم my sql

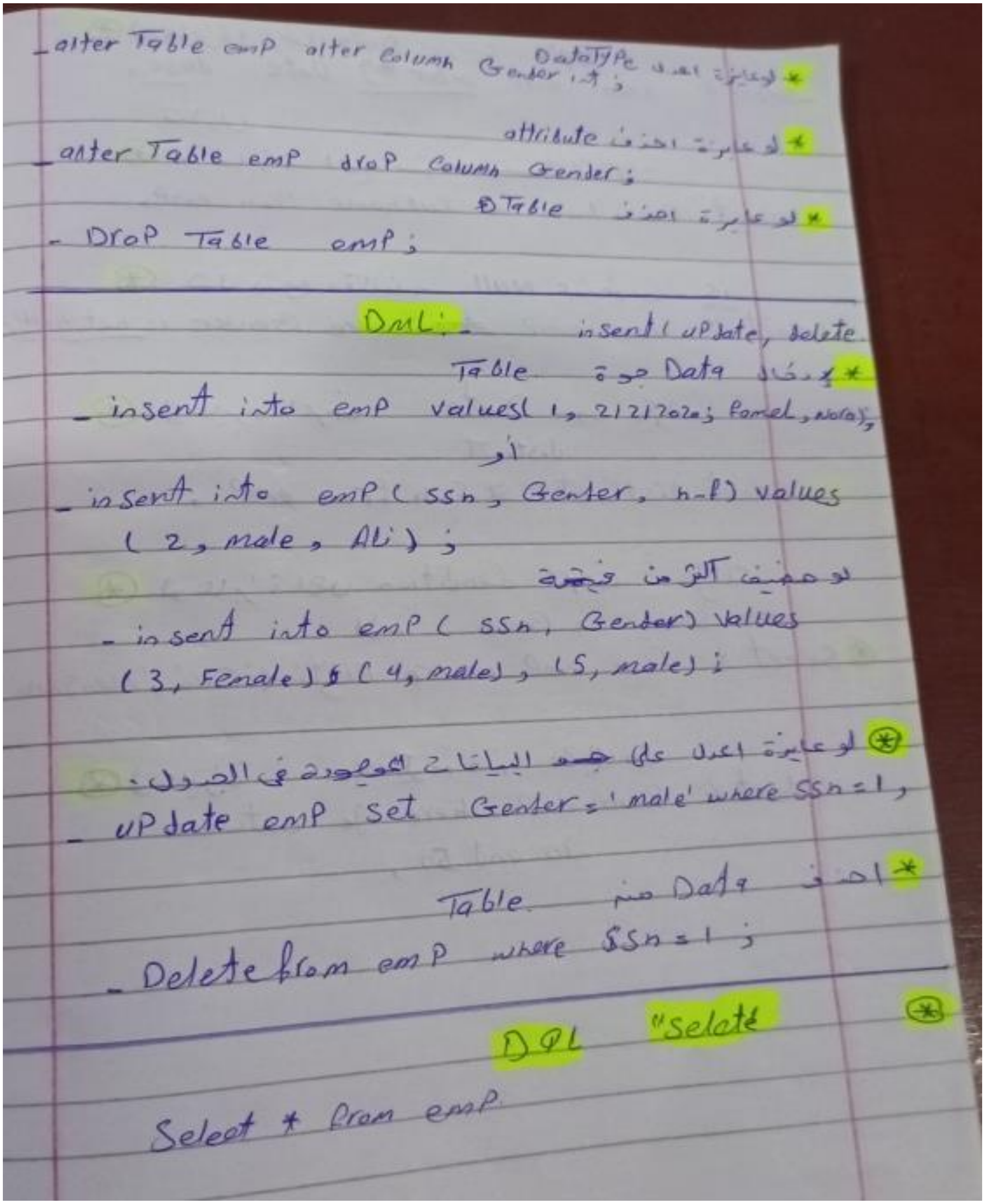
* DDL → Data Definition lang
* DML → Data manipulation lang.
* DCL → Data Control lang
query lang

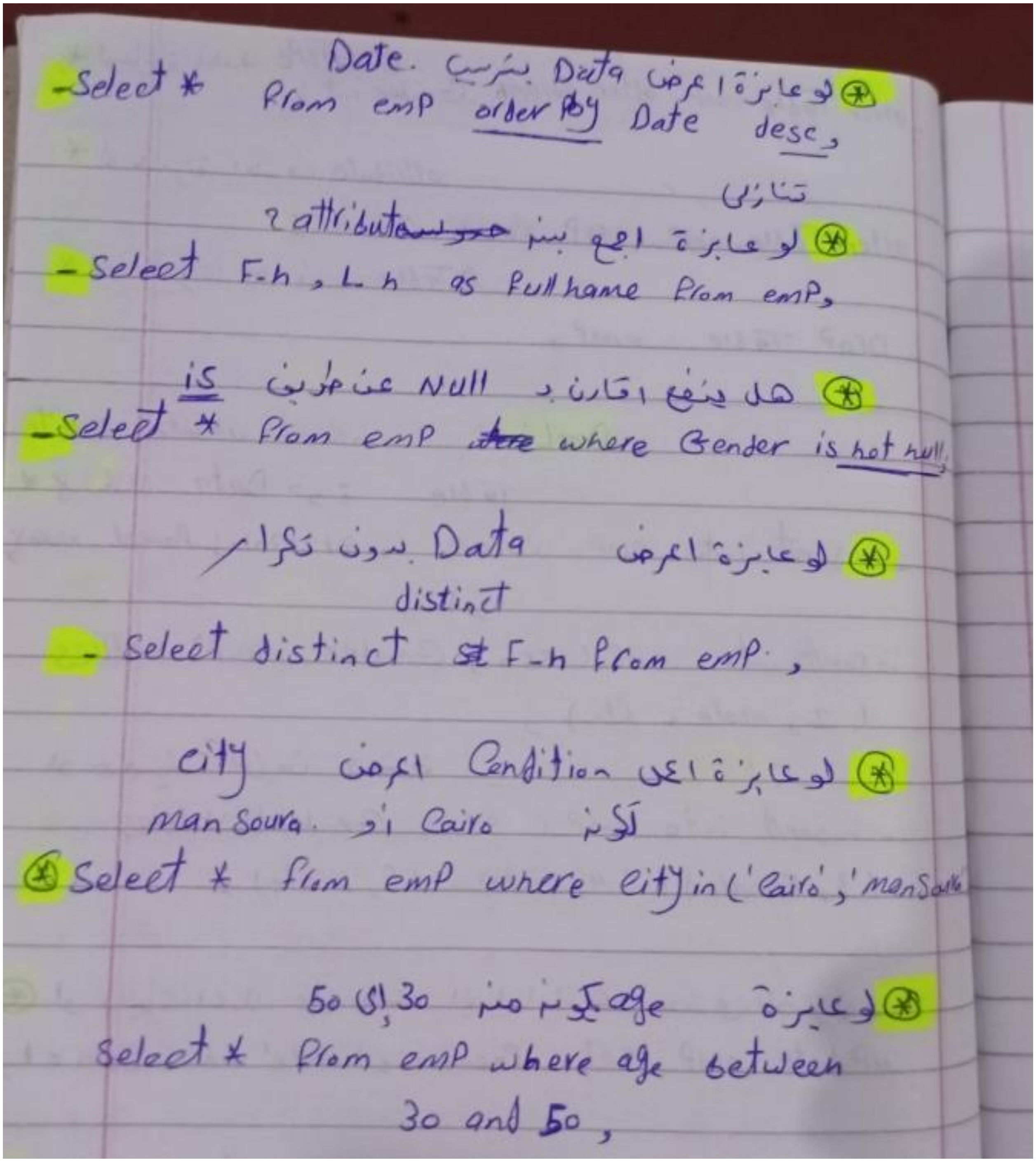


```
alter table emp add sal int

alter table emp alter column sal bigint

alter table emp drop column sal


drop table emp
```



```
--DML
--insert update delete
insert into emp
values(1,'ali',NULL,'alex','1/1/2010',NULL)

insert into emp(ename,eid)
values('eman',9)

--insert constructor
insert into emp(ename,eid)
values('eman',8),('ali',12),('nada',7)
```





⊛ لو عايزة اعرض Data بترتيب Date.

-Select * from emp order by Date desc,

تنازلى

⊛ لو عايزة اجمع بين ~~حرس~~ 2 attribute

-select F-n , L-n as fullname from emp,

⊛ هل ينفع اقارن بـ Null عن طريق is

-select * from emp ~~there~~ where Gender is not null,

⊛ لو عايزة اعرض Data بدون تكرار /

distinct

- select distinct ~~st~~ F-n from emp ,

⊛ لو عايزة اعمل Condition اعرض city

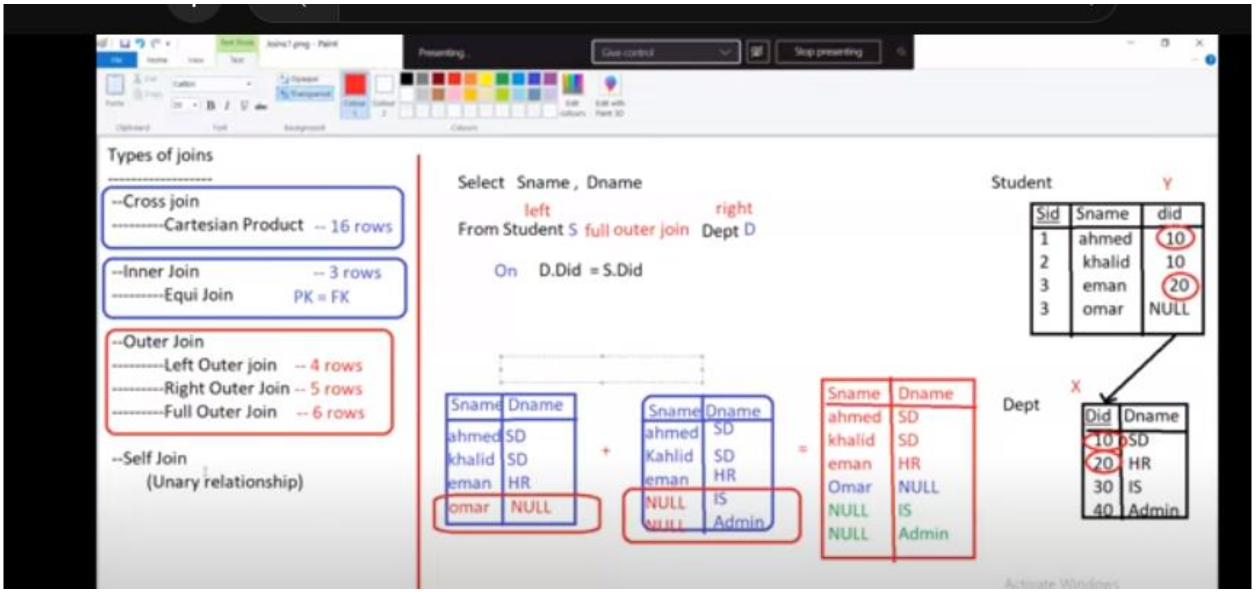
تكون Cairo أو mansoura.

⊛ Select * from emp where city in ('Cairo','mansoura

⊛ لو عايزة age يكون من 30 إلى 50

Select * from emp where age between 30 and 50 ,

3-SQL Joins, Normalization



Types of joins

--Cross join
----------Cartesian Product -- 16 rows

--Inner Join -- 3 rows
----------Equi Join PK = FK

--Outer Join
----------Left Outer join -- 4 rows
----------Right Outer Join -- 5 rows
----------Full Outer Join -- 6 rows

--Self Join
(Unary relationship)

```
Select  Sname , Dname
                   left                    right
From Student S  full outer join  Dept D

     On   D.Did  = S.Did
```

Student Y

Sid	Sname	did
1	ahmed	10
2	khalid	10
3	eman	20
3	omar	NULL

Dept X

Did	Dname
10	SD
20	HR
30	IS
40	Admin

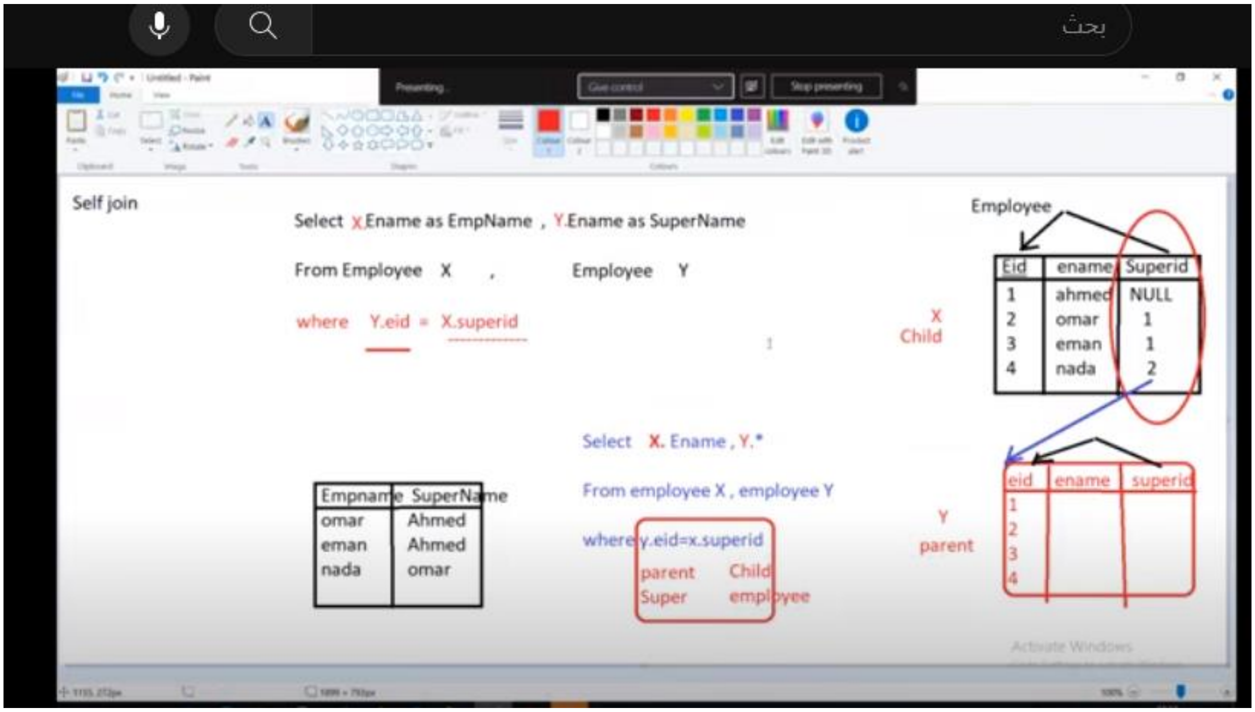
Sname	Dname
ahmed	SD
khalid	SD
eman	HR
omar	NULL

+

Sname	Dname
ahmed	SD
Kahlid	SD
eman	HR
NULL	IS
NULL	Admin

=

Sname	Dname
ahmed	SD
khalid	SD
eman	HR
Omar	NULL
NULL	IS
NULL	Admin



Self join

```
Select  X.Ename as EmpName , Y.Ename as SuperName

From Employee   X    ,        Employee   Y

where   Y.eid  =  X.superid
```

Employee

Eid	ename	Superid
1	ahmed	NULL
2	omar	1
3	eman	1
4	nada	2

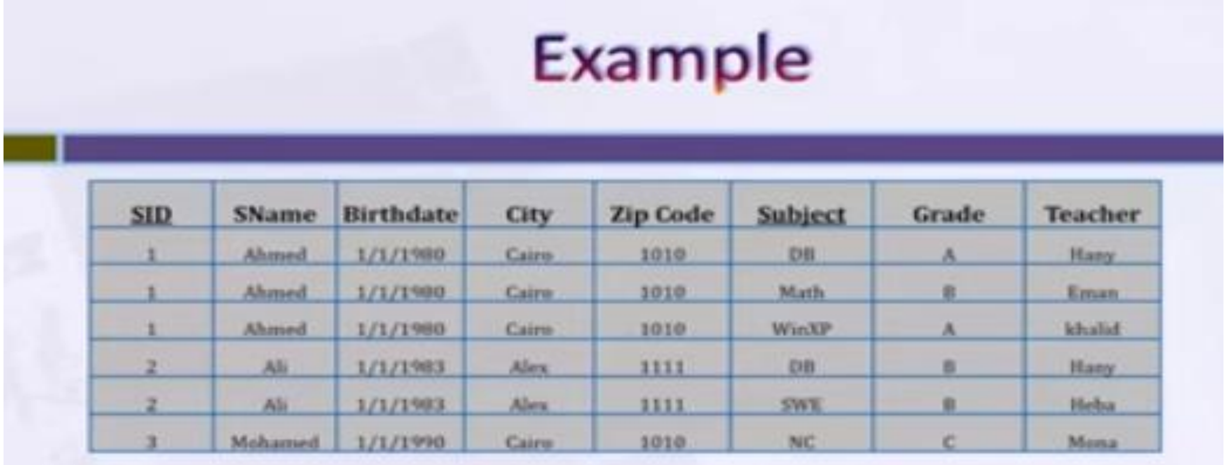
X
Child

```
Select   X. Ename , Y.*

From employee X , employee Y

where y.eid=x.superid
      parent    Child
      Super     employee
```

Y
parent

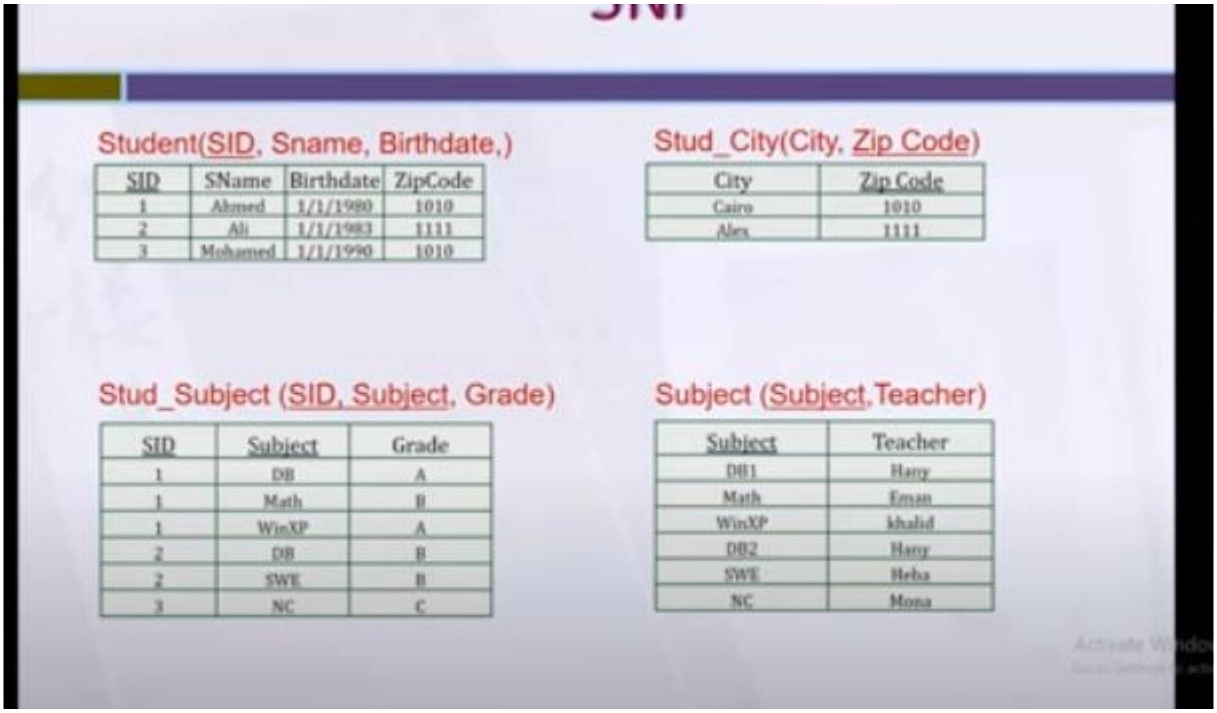
eid	ename	superid
1		
2		
3		
4		

Empname	SuperName
omar	Ahmed
eman	Ahmed
nada	omar



Example

SID	SName	Birthdate	City	Zip Code	Subject	Grade	Teacher
1	Ahmed	1/1/1980	Cairo	1010	DB	A	Hany
1	Ahmed	1/1/1980	Cairo	1010	Math	B	Eman
1	Ahmed	1/1/1980	Cairo	1010	WinXP	A	khalid
2	Ali	1/1/1983	Alex	1111	DB	B	Hany
2	Ali	1/1/1983	Alex	1111	SWE	B	Heba
3	Mohamed	1/1/1990	Cairo	1010	NC	C	Mona



3NF

Student(SID, Sname, Birthdate,)

SID	SName	Birthdate	ZipCode
1	Ahmed	1/1/1980	1010
2	Ali	1/1/1983	1111
3	Mohamed	1/1/1990	1010

Stud_City(City, Zip Code)

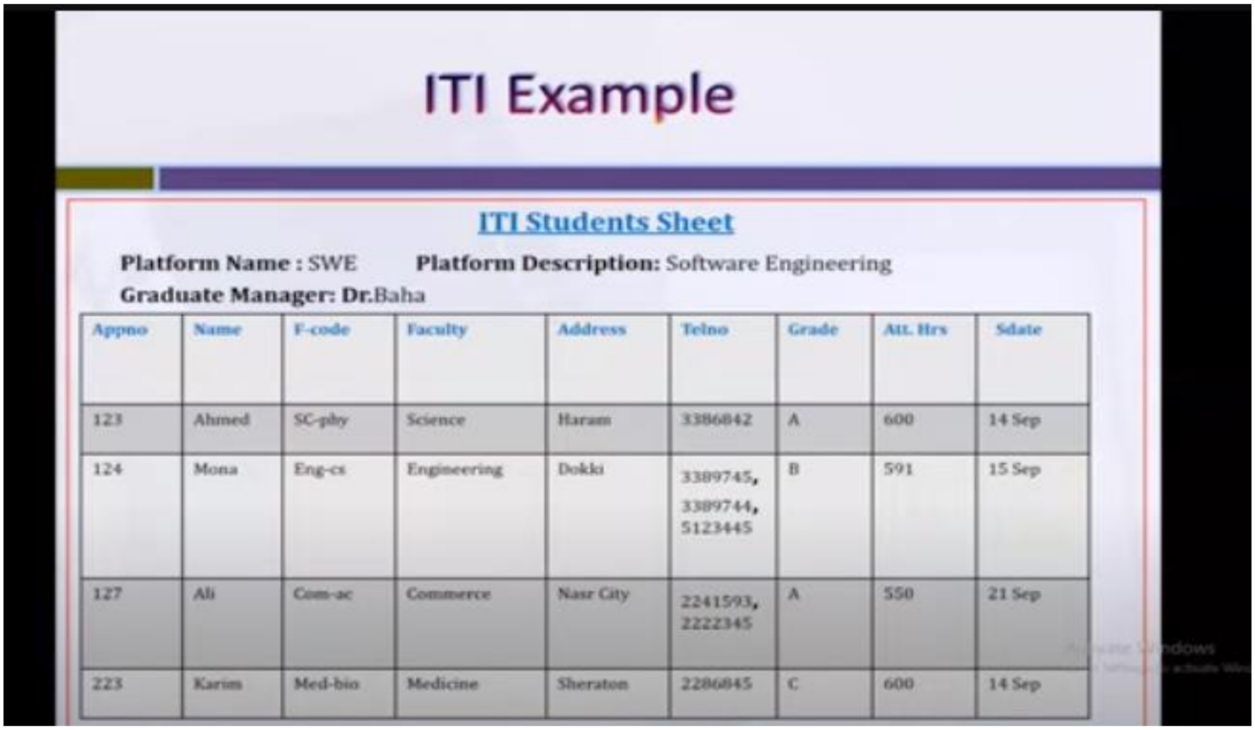
City	Zip Code
Cairo	1010
Alex	1111

Stud_Subject (SID, Subject, Grade)

SID	Subject	Grade
1	DB	A
1	Math	B
1	WinXP	A
2	DB	B
2	SWE	B
3	NC	C

Subject (Subject,Teacher)

Subject	Teacher
DB1	Hany
Math	Eman
WinXP	khalid
DB2	Hany
SWE	Heba
NC	Mona



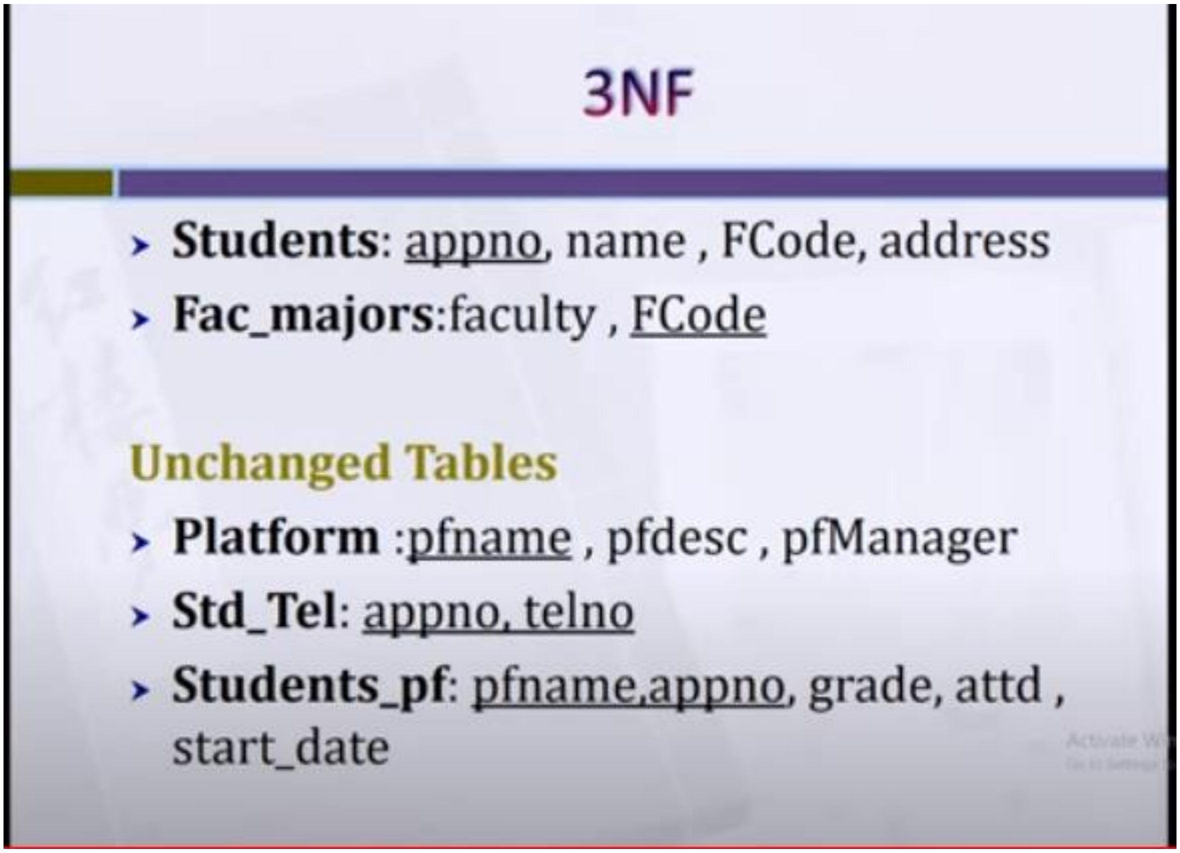
ITI Example

ITI Students Sheet

Platform Name : SWE **Platform Description:** Software Engineering

Graduate Manager: Dr. Baha

Appno	Name	F-code	Faculty	Address	Telno	Grade	Att. Hrs	Sdate
123	Ahmed	SC-phy	Science	Haram	3386842	A	600	14 Sep
124	Mona	Eng-cs	Engineering	Dokki	3389745, 3389744, 5123445	B	591	15 Sep
127	Ali	Com-ac	Commerce	Nasr City	2241593, 2222345	A	550	21 Sep
223	Karim	Med-bio	Medicine	Sheraton	2286845	C	600	14 Sep



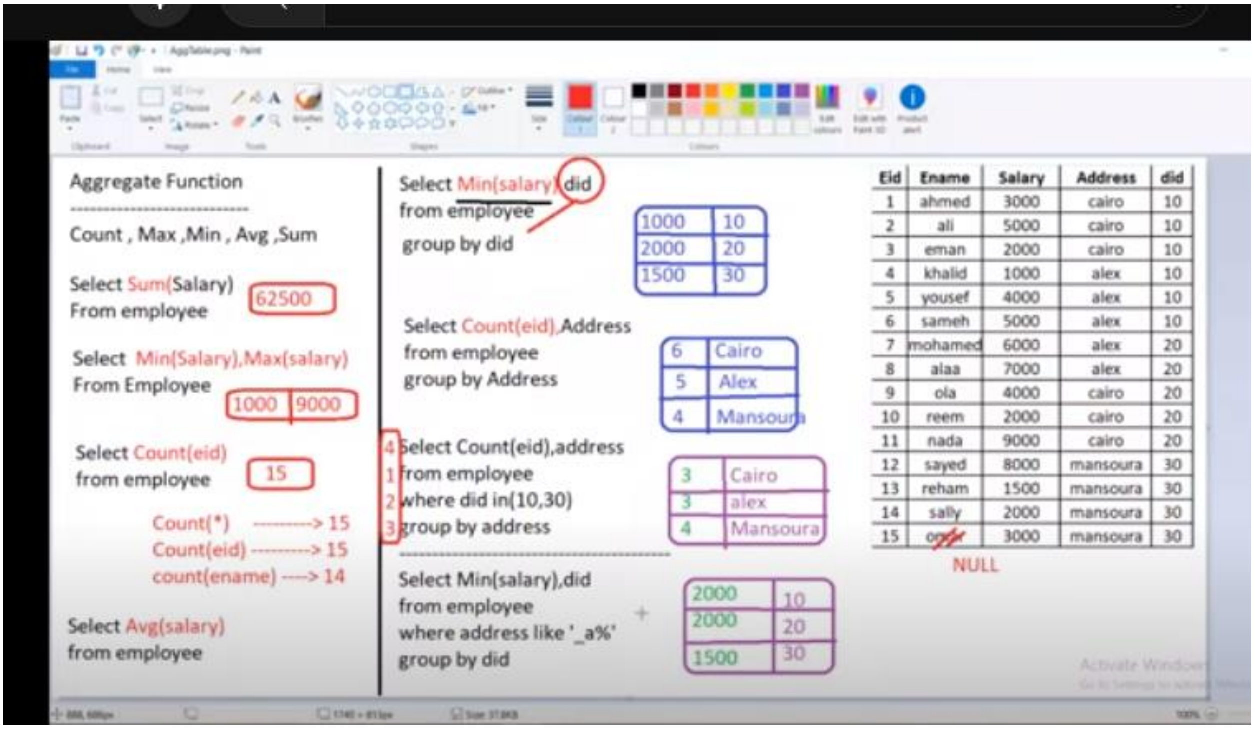
3NF

- **Students**: appno, name , FCode, address
- **Fac_majors**:faculty , FCode

Unchanged Tables

- **Platform** :pfname , pfdesc , pfManager
- **Std_Tel**: appno, telno
- **Students_pf**: pfname,appno, grade, attd , start_date

4-SQL, Aggregate Function, Grouping, Union, Subqueries, EERD



Aggregate Function

Count , Max ,Min , Avg ,Sum

```
Select Sum(Salary)
From employee
```

62500

```
Select  Min(Salary),Max(salary)
From Employee
```

1000	9000

```
Select Count(eid)
from employee
```

15

```
Count(*)     ---------> 15
Count(eid) ---------> 15
count(ename) ----> 14
```

```
Select Avg(salary)
from employee
```

```
Select Min(salary),did
from employee
group by did
```

1000	10
2000	20
1500	30

```
Select Count(eid),Address
from employee
group by Address
```

6	Cairo
5	Alex
4	Mansoura

```
4 Select Count(eid),address
1 from employee
2 where did in(10,30)
3 group by address
```

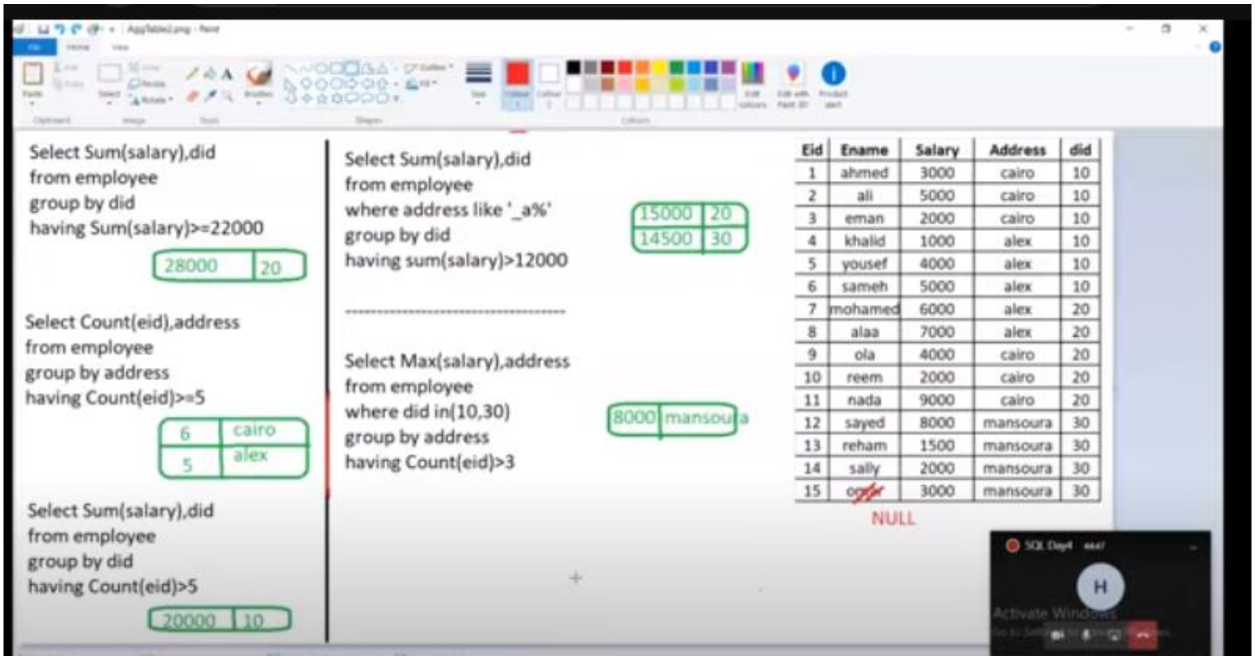
3	Cairo
3	alex
4	Mansoura

```
Select Min(salary),did
from employee
where address like '_a%'
group by did
```

2000	10
2000	20
1500	30

Eid	Ename	Salary	Address	did
1	ahmed	3000	cairo	10
2	ali	5000	cairo	10
3	eman	2000	cairo	10
4	khalid	1000	alex	10
5	yousef	4000	alex	10
6	sameh	5000	alex	10
7	mohamed	6000	alex	20
8	alaa	7000	alex	20
9	ola	4000	cairo	20
10	reem	2000	cairo	20
11	nada	9000	cairo	20
12	sayed	8000	mansoura	30
13	reham	1500	mansoura	30
14	sally	2000	mansoura	30
15	~~omar~~	3000	mansoura	30

NULL



```
Select Sum(salary),did
from employee
group by did
having Sum(salary)>=22000
```

28000	20

```
Select Count(eid),address
from employee
group by address
having Count(eid)>=5
```

6	cairo
5	alex

```
Select Sum(salary),did
from employee
group by did
having Count(eid)>5
```

20000	10

```
Select Sum(salary),did
from employee
where address like '_a%'
group by did
having sum(salary)>12000
```

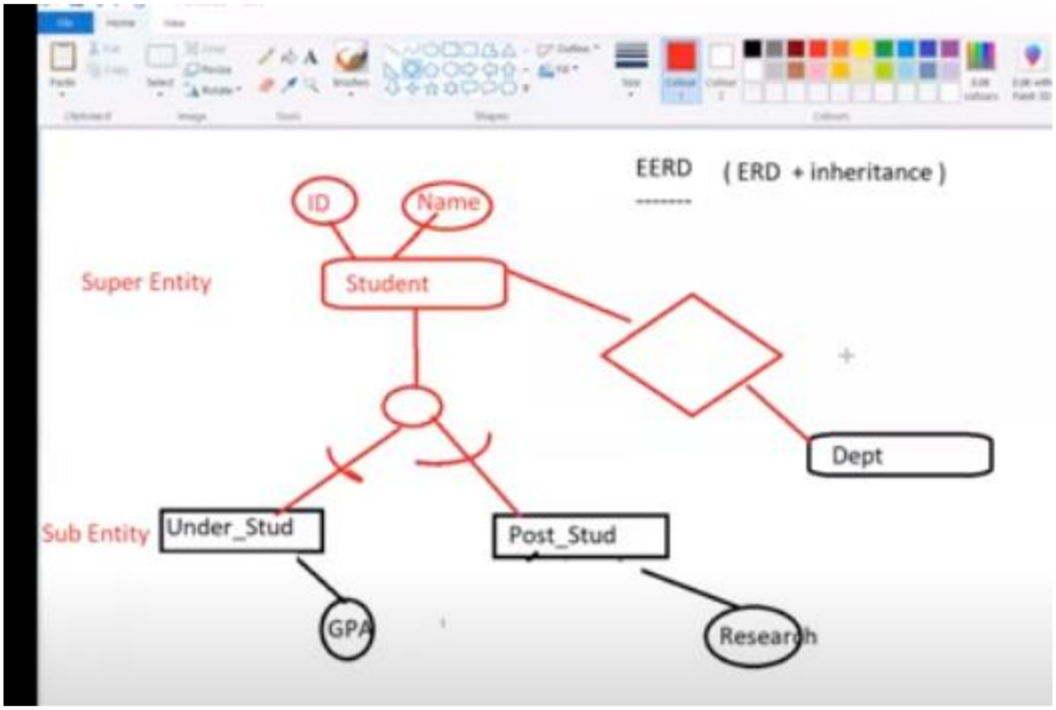
15000	20
14500	30

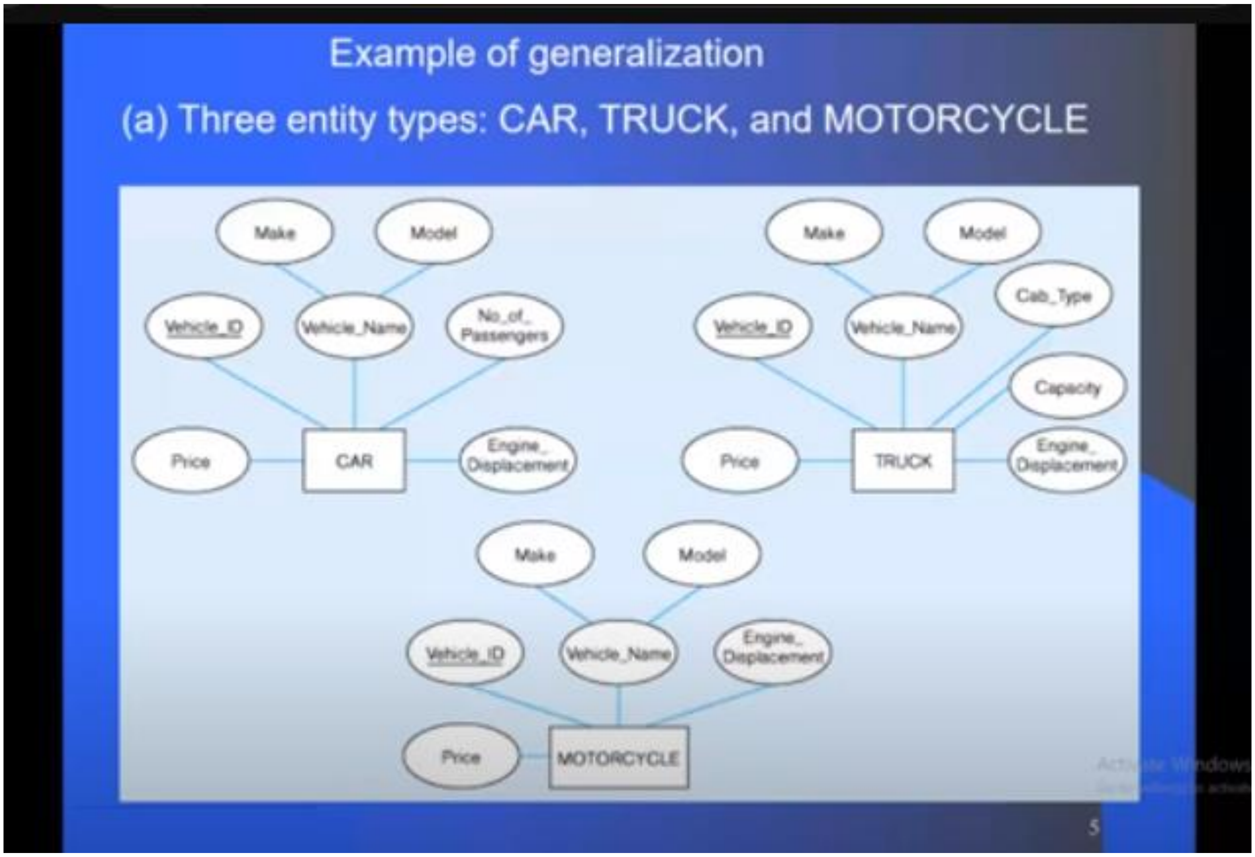
```
Select Max(salary),address
from employee
where did in(10,30)
group by address
having Count(eid)>3
```

8000	mansoura

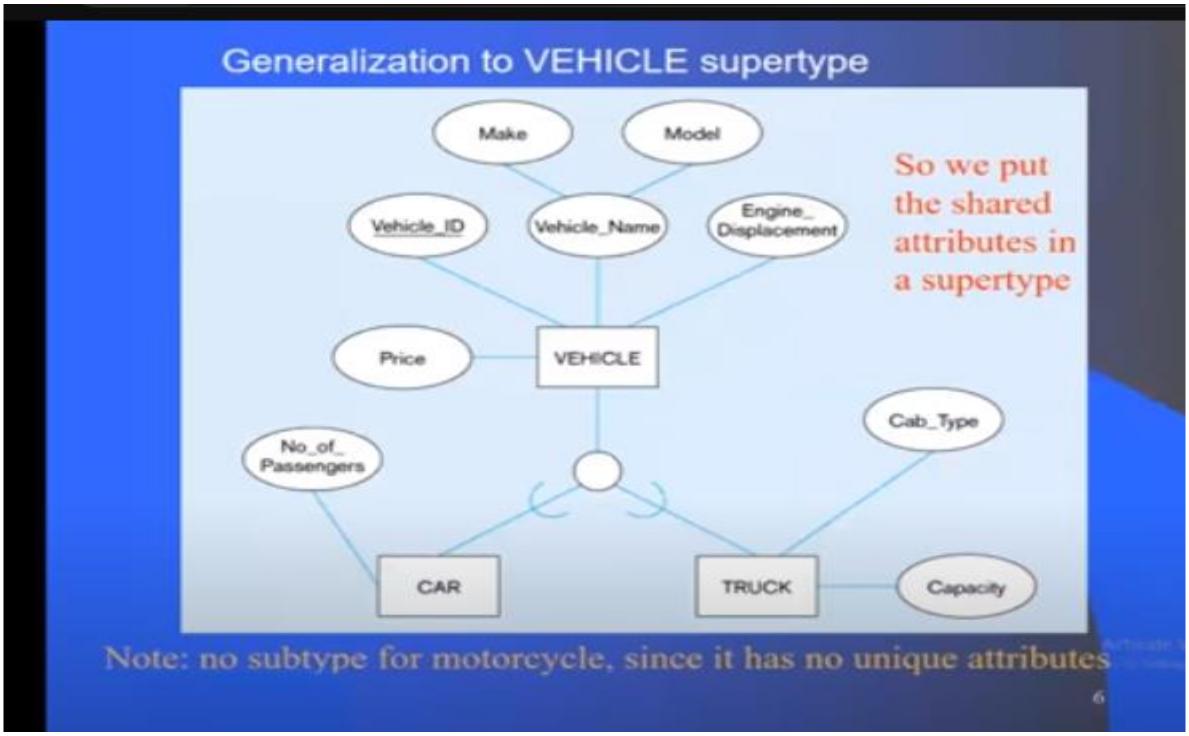
Eid	Ename	Salary	Address	did
1	ahmed	3000	cairo	10
2	ali	5000	cairo	10
3	eman	2000	cairo	10
4	khalid	1000	alex	10
5	yousef	4000	alex	10
6	sameh	5000	alex	10
7	mohamed	6000	alex	20
8	alaa	7000	alex	20
9	ola	4000	cairo	20
10	reem	2000	cairo	20
11	nada	9000	cairo	20
12	sayed	8000	mansoura	30
13	reham	1500	mansoura	30
14	sally	2000	mansoura	30
15	~~omar~~	3000	mansoura	30

NULL




Example of generalization

(a) Three entity types: CAR, TRUCK, and MOTORCYCLE

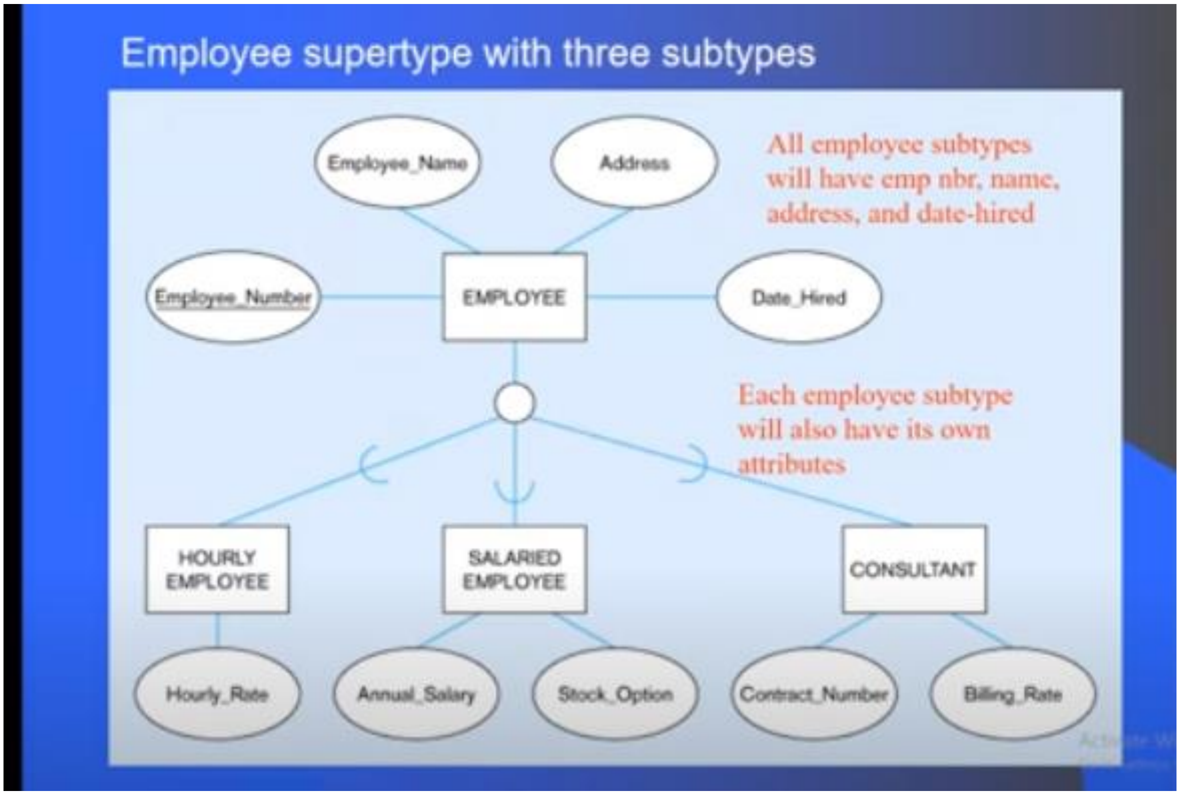


Generalization to VEHICLE supertype

So we put the shared attributes in a supertype

Note: no subtype for motorcycle, since it has no unique attributes

New example

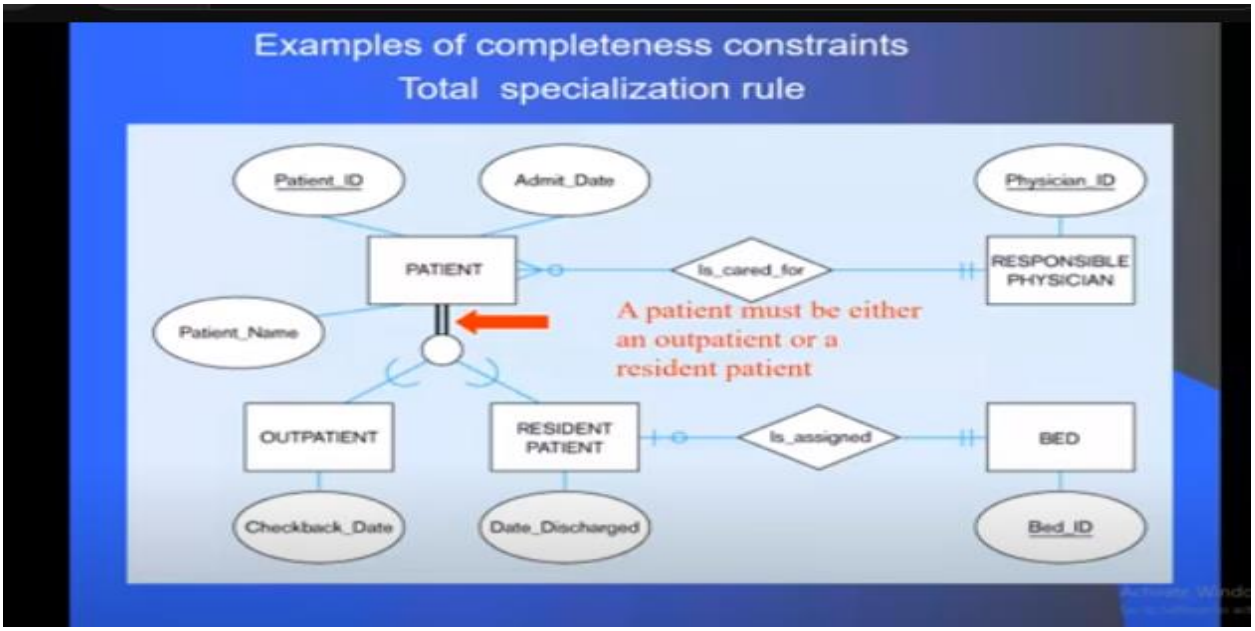


Employee supertype with three subtypes

All employee subtypes will have emp nbr, name, address, and date-hired

Each employee subtype will also have its own attributes

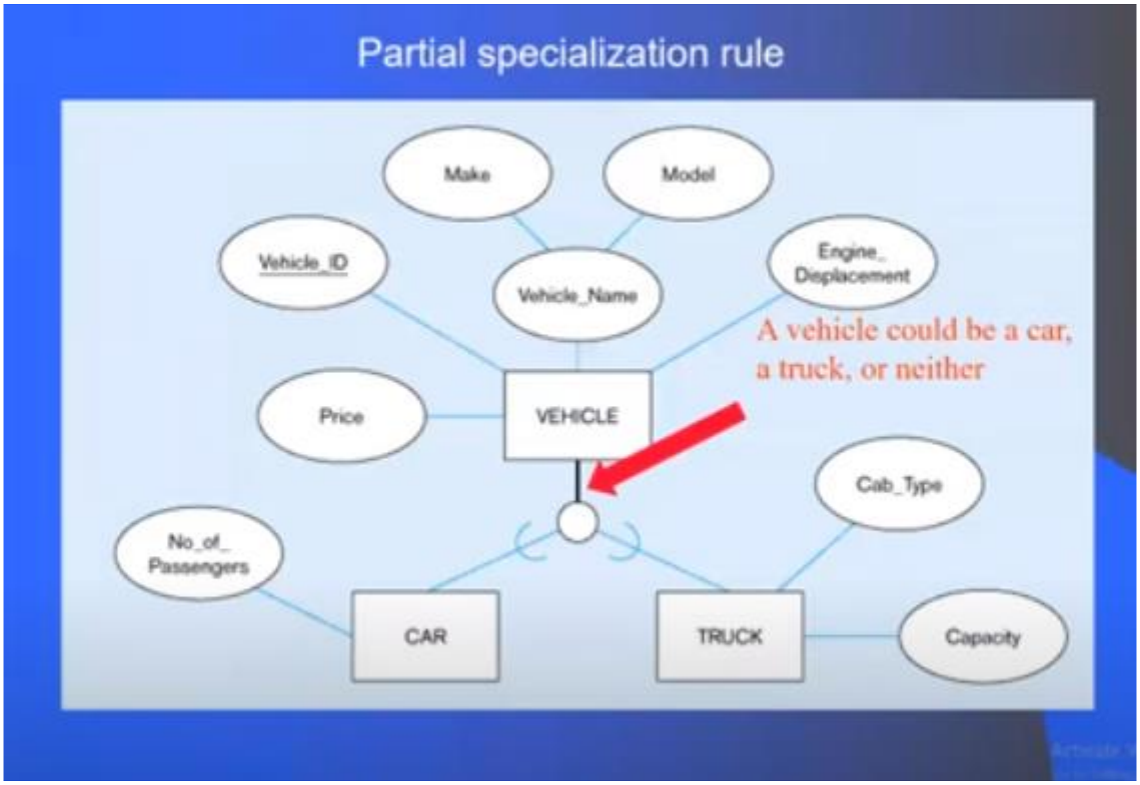
Total



Examples of completeness constraints
Total specialization rule

A patient must be either an outpatient or a resident patient

partial

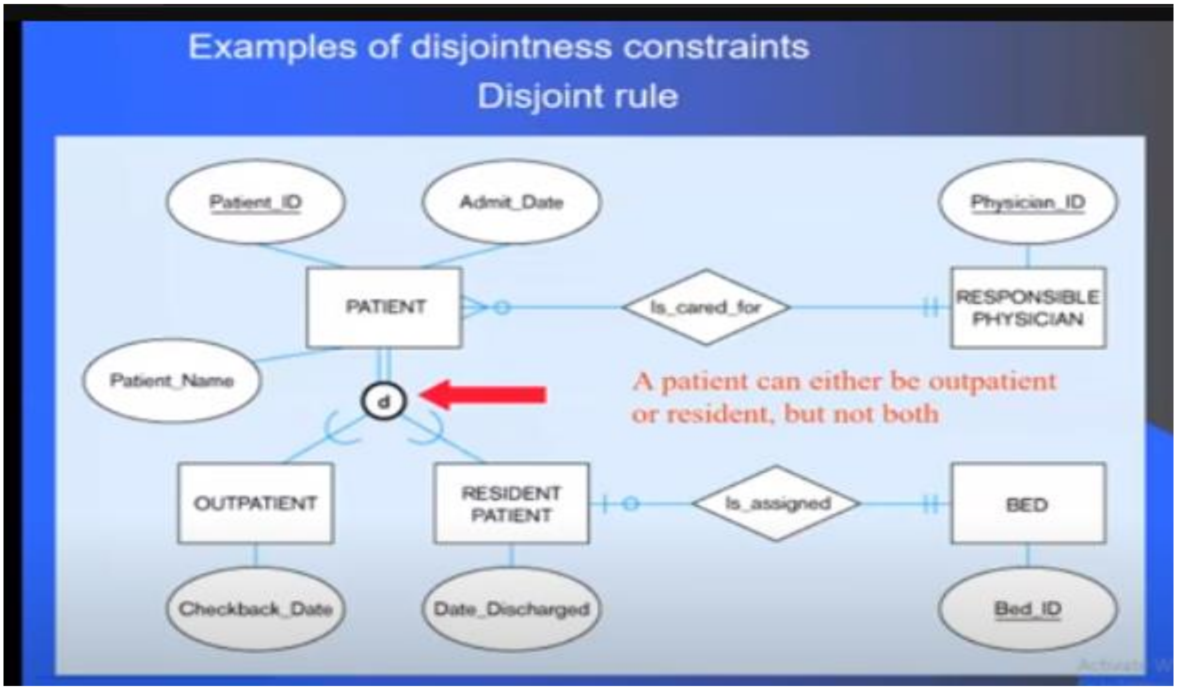


Partial specialization rule

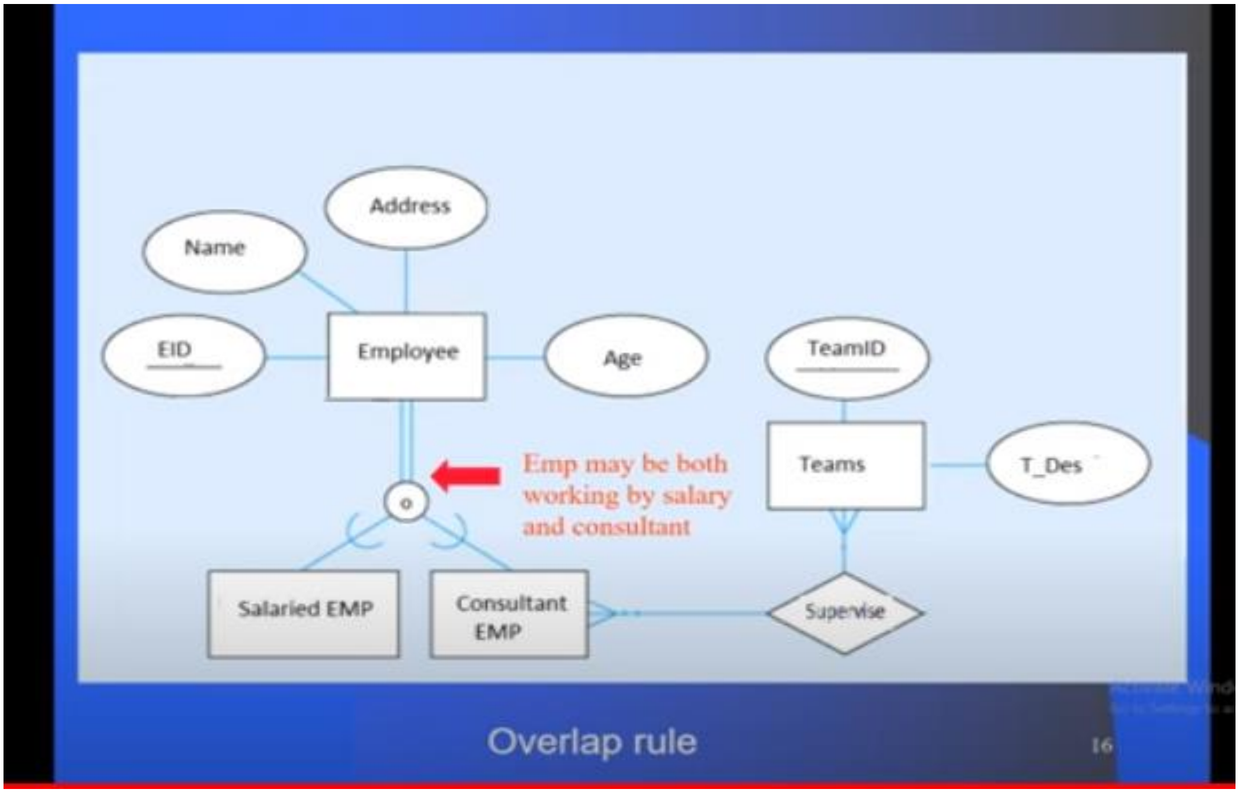
A vehicle could be a car, a truck, or neither

Examples of disjointness constraints

Disjoint rule



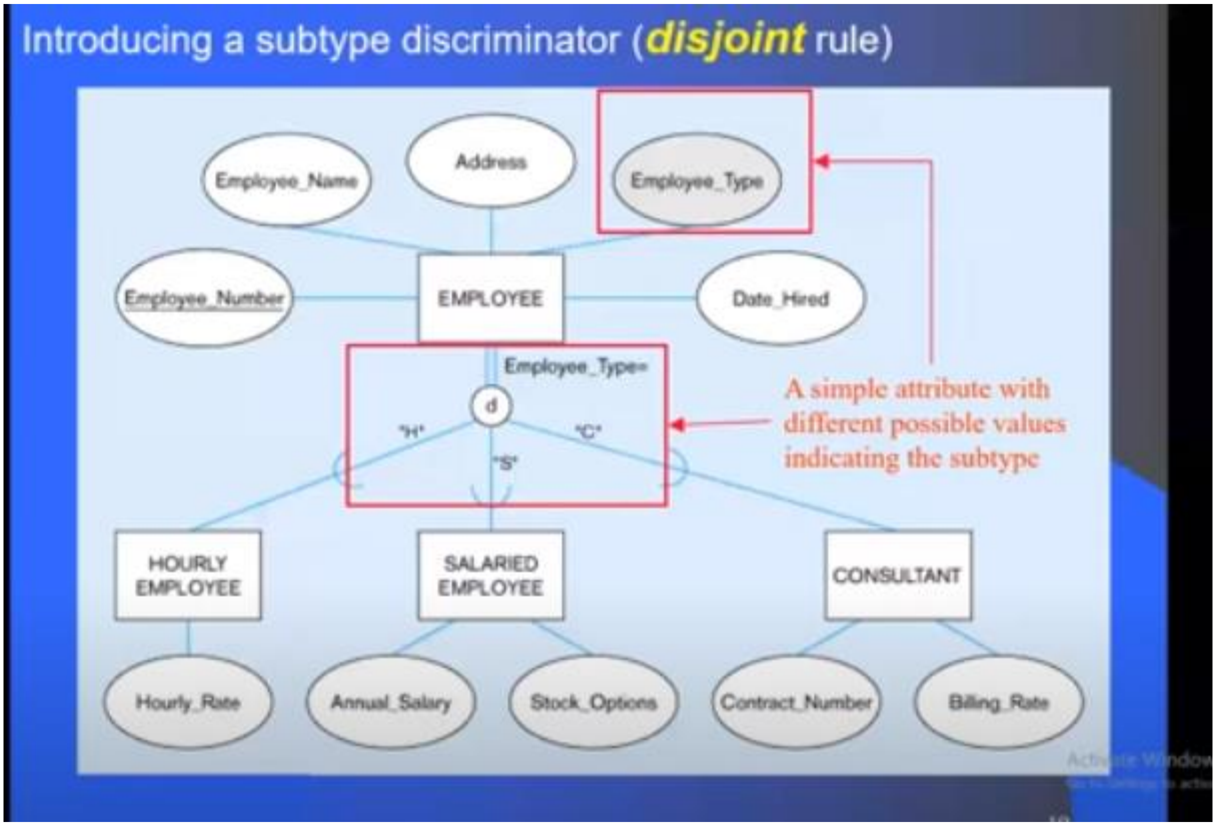
A patient can either be outpatient or resident, but not both



Emp may be both working by salary and consultant

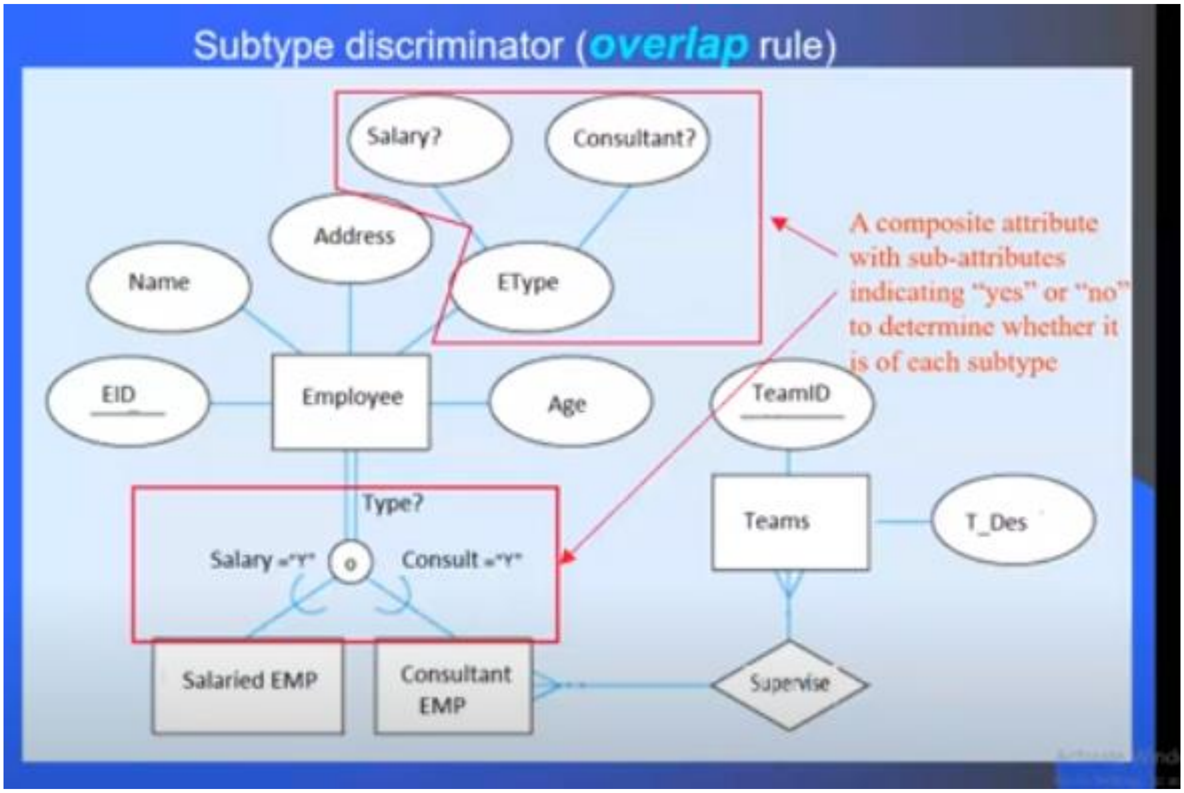
Overlap rule

16



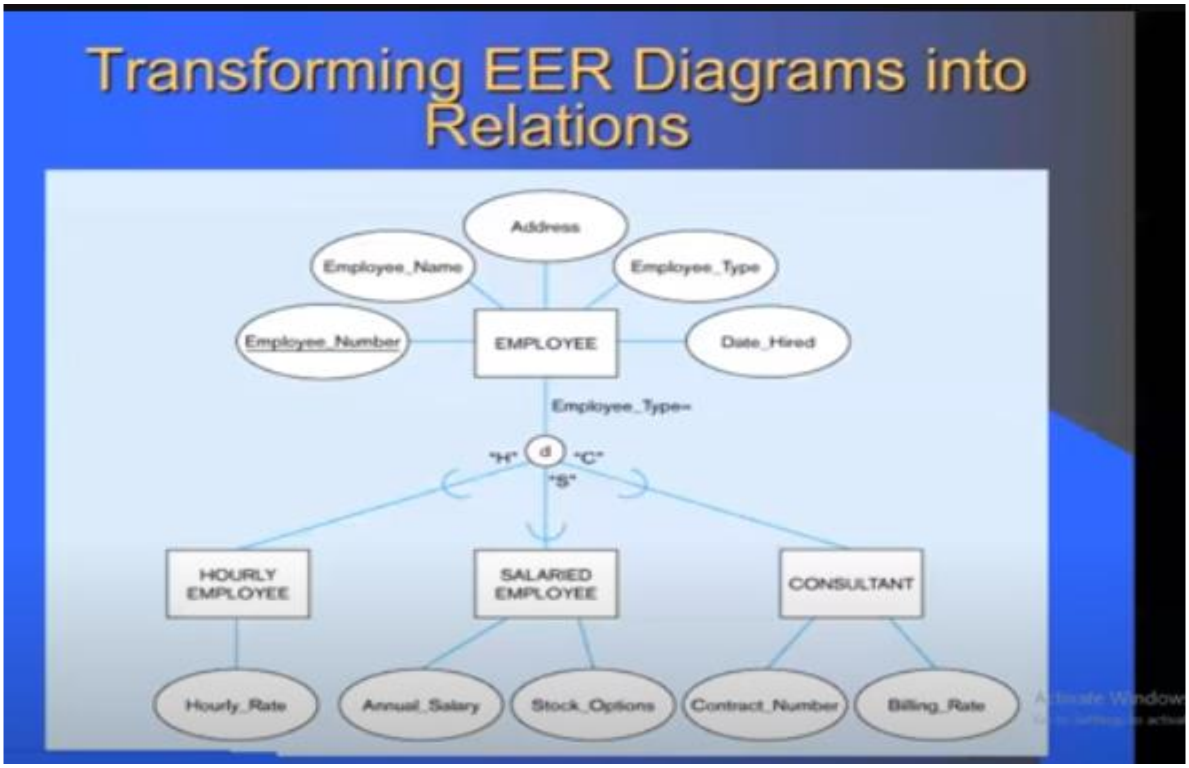
Introducing a subtype discriminator (*disjoint* rule)

A simple attribute with different possible values indicating the subtype

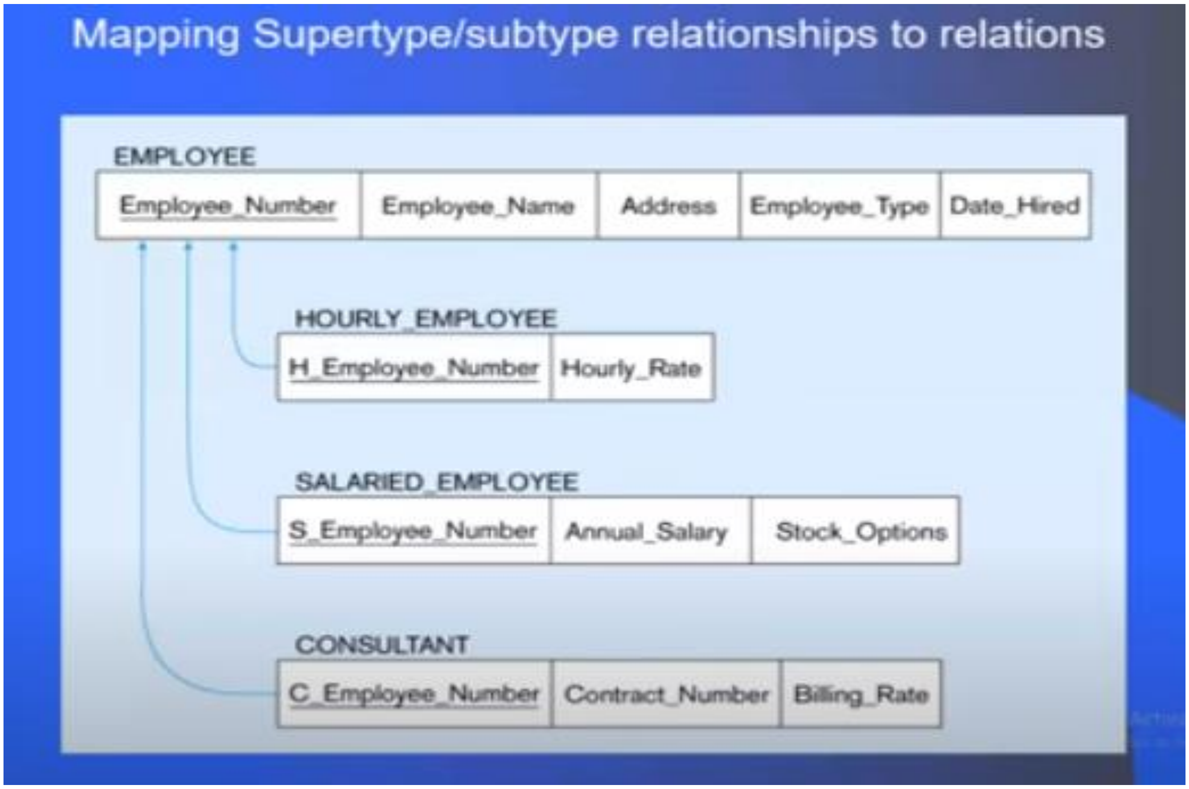


Subtype discriminator (*overlap* rule)

A composite attribute with sub-attributes indicating "yes" or "no" to determine whether it is of each subtype



Transforming EER Diagrams into Relations



Mapping Supertype/subtype relationships to relations

EMPLOYEE

Employee_Number	Employee_Name	Address	Employee_Type	Date_Hired

HOURLY_EMPLOYEE

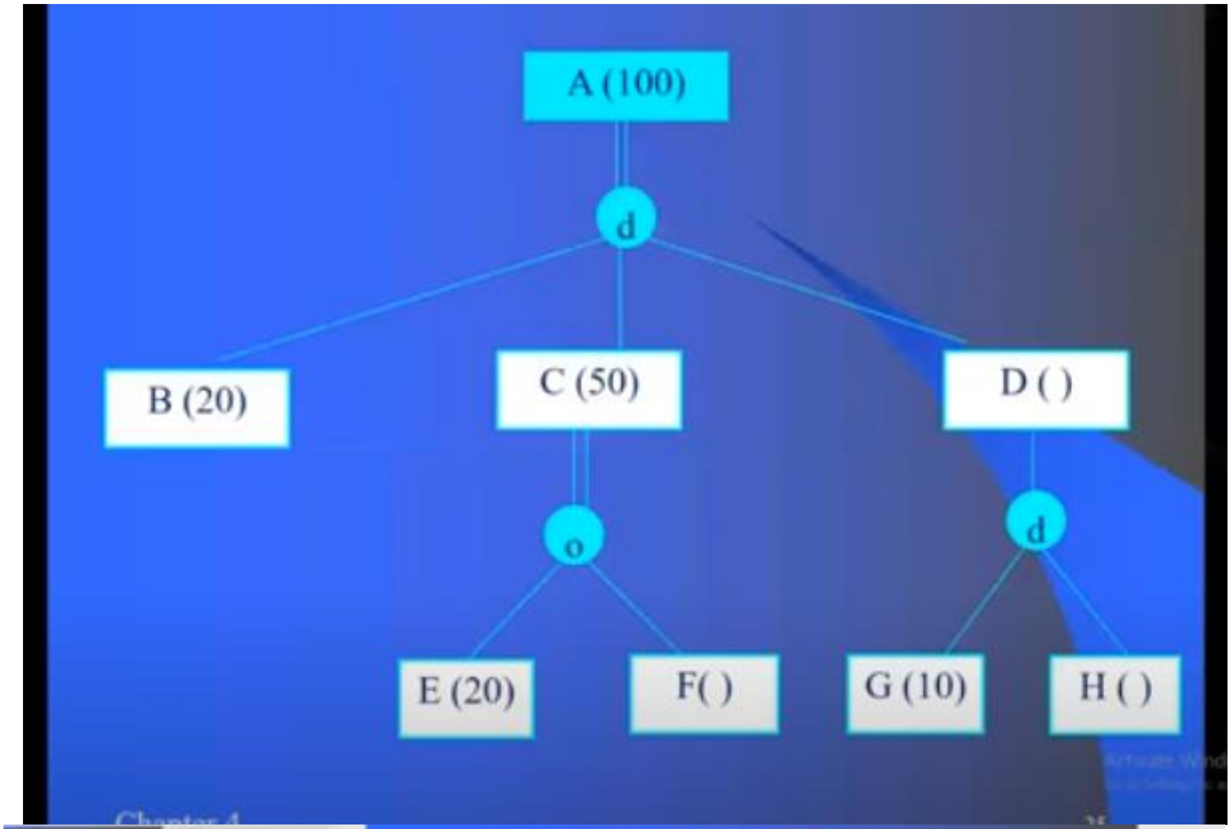
H_Employee_Number	Hourly_Rate

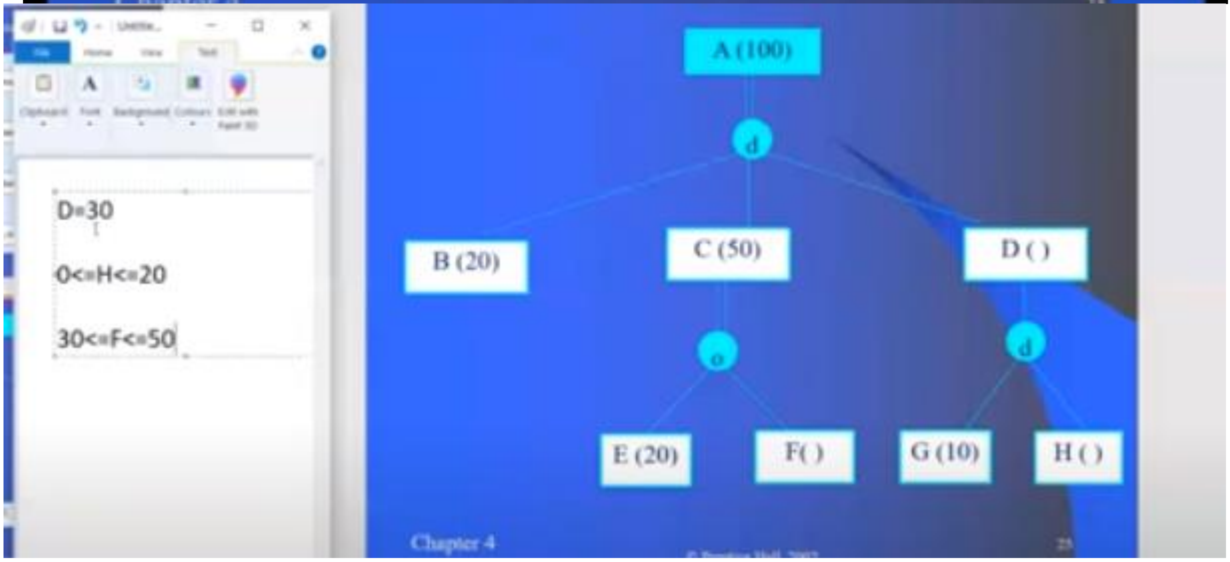
SALARIED_EMPLOYEE

S_Employee_Number	Annual_Salary	Stock_Options

CONSULTANT

C_Employee_Number	Contract_Number	Billing_Rate

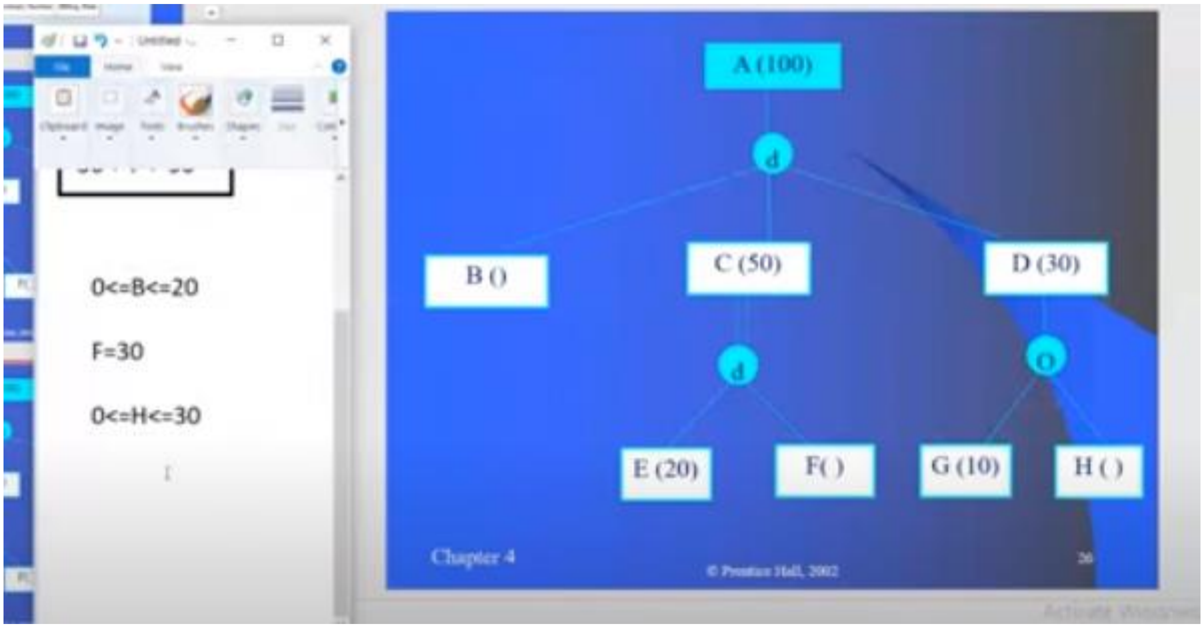




D=30

0<=H<=20

30<=F<=50



0<=B<=20

F=30

0<=H<=30

A (100)

d

B ()

C (50)

D (30)

d

o

E (20)

F( )

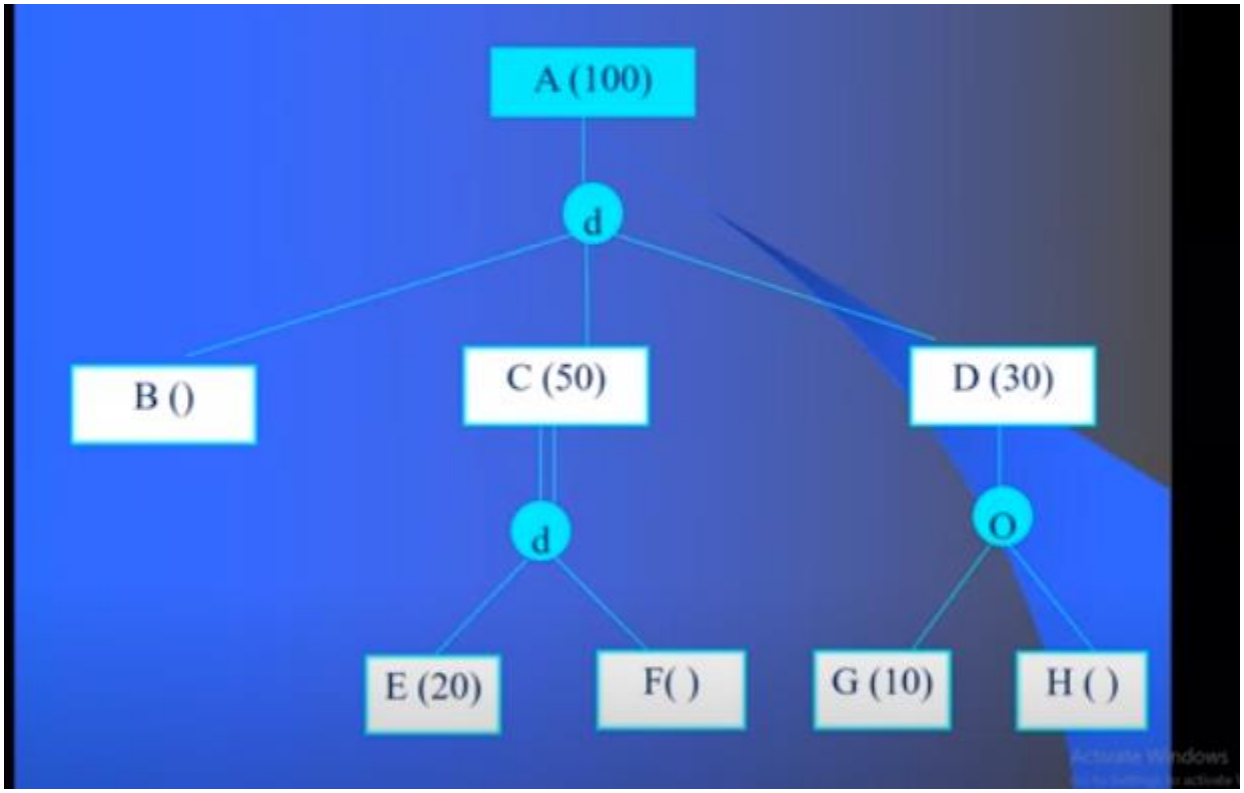
G (10)

H ( )

Chapter 4

© Prentice Hall, 2002

26



A (100)

d

B ()

C (50)

D (30)

d

O

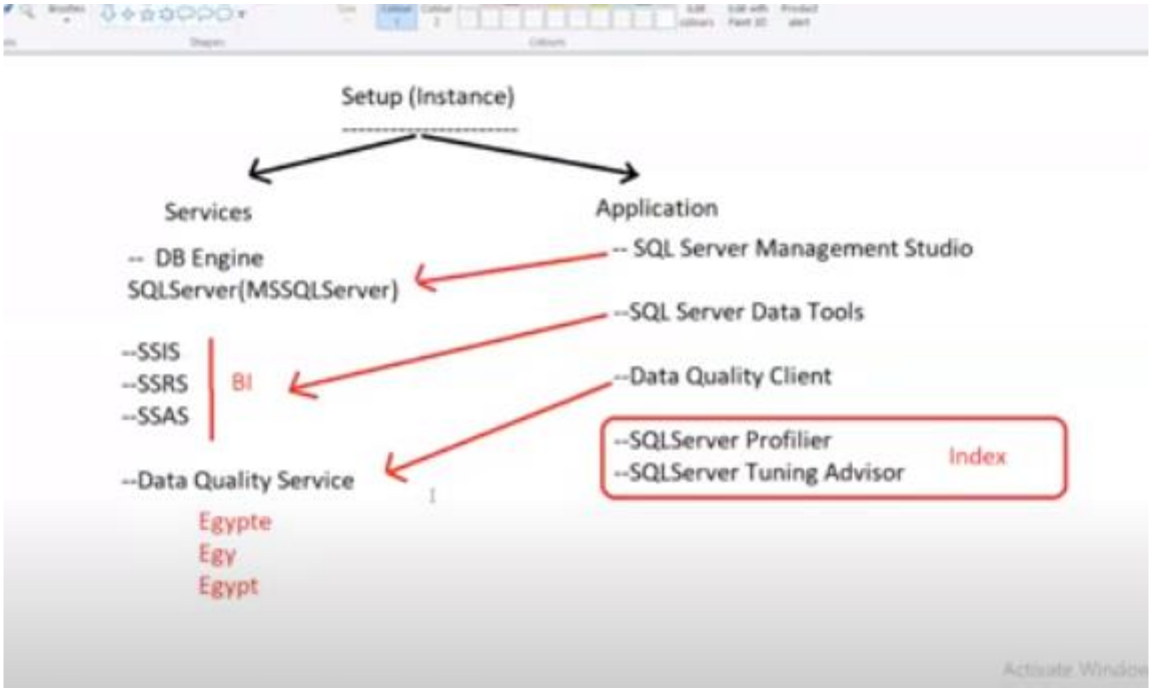
E (20)

F( )

G (10)

H ( )

5-SQL, DB Engine, SQL Services, Ranking Function, Transact SQL



Setup (Instance)

Services

-- DB Engine
SQLServer(MSSQLServer)

--SSIS
--SSRS
--SSAS

BI

--Data Quality Service

Egypte
Egy
Egypt

Application

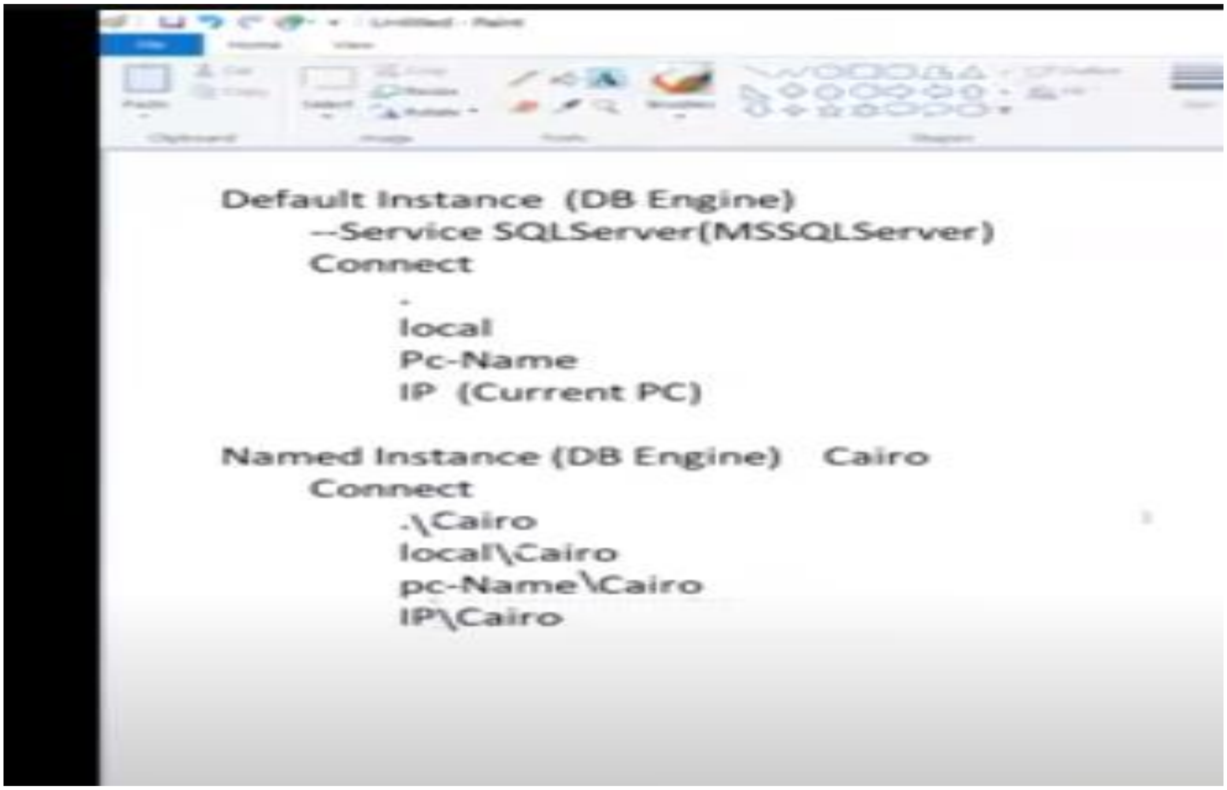
-- SQL Server Management Studio

--SQL Server Data Tools

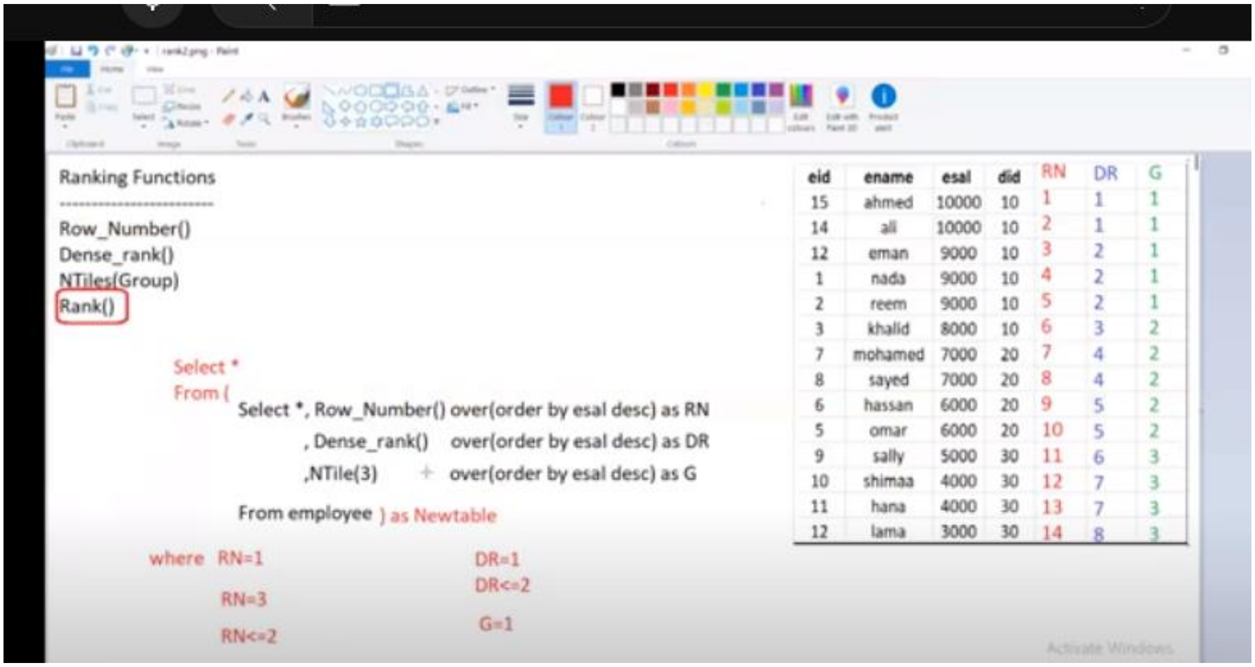
--Data Quality Client

--SQLServer Profilier
--SQLServer Tuning Advisor

Index



```
Default Instance (DB Engine)
        --Service SQLServer(MSSQLServer)
        Connect
                .
                local
                Pc-Name
                IP (Current PC)

Named Instance (DB Engine)    Cairo
        Connect
                .\Cairo
                local\Cairo
                pc-Name\Cairo
                IP\Cairo
```



Ranking Functions

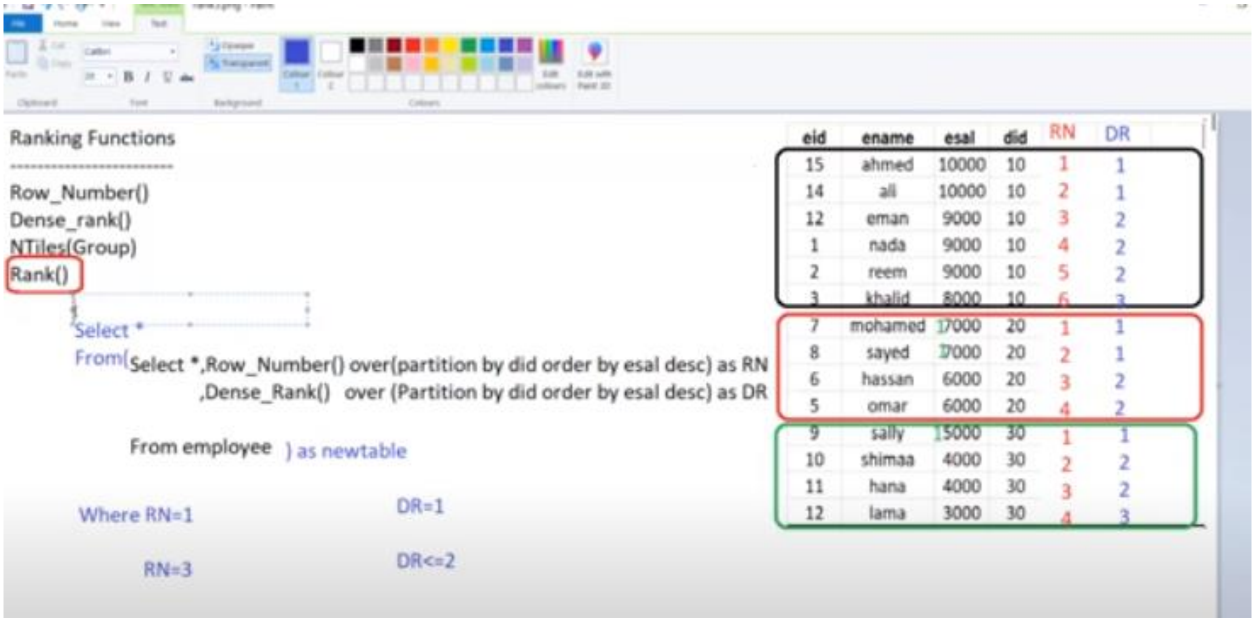
-----------------------

Row_Number()
Dense_rank()
NTiles(Group)
Rank()

```
Select *
From (
      Select *, Row_Number() over(order by esal desc) as RN
              , Dense_rank()    over(order by esal desc) as DR
              ,NTile(3)         over(order by esal desc) as G
      From employee ) as Newtable

where RN=1                         DR=1
                                   DR<=2
      RN=3
                                   G=1
      RN<=2
```

eid	ename	esal	did	RN	DR	G
15	ahmed	10000	10	1	1	1
14	ali	10000	10	2	1	1
12	eman	9000	10	3	2	1
1	nada	9000	10	4	2	1
2	reem	9000	10	5	2	1
3	khalid	8000	10	6	3	2
7	mohamed	7000	20	7	4	2
8	sayed	7000	20	8	4	2
6	hassan	6000	20	9	5	2
5	omar	6000	20	10	5	2
9	sally	5000	30	11	6	3
10	shimaa	4000	30	12	7	3
11	hana	4000	30	13	7	3
12	lama	3000	30	14	8	3

Activate Windows



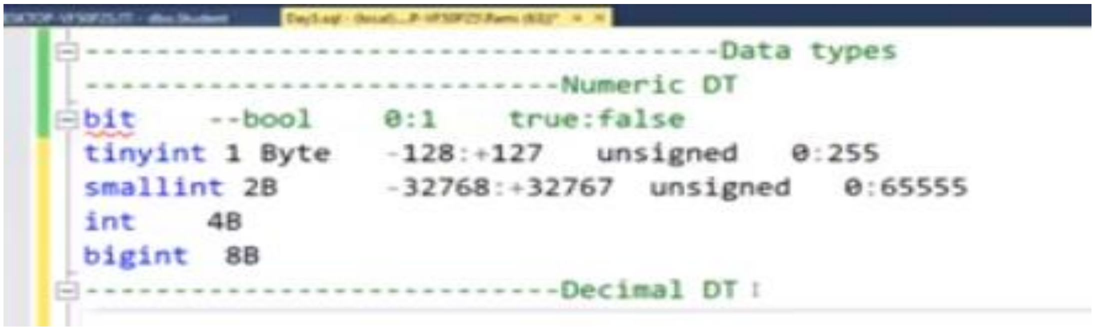
Ranking Functions

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

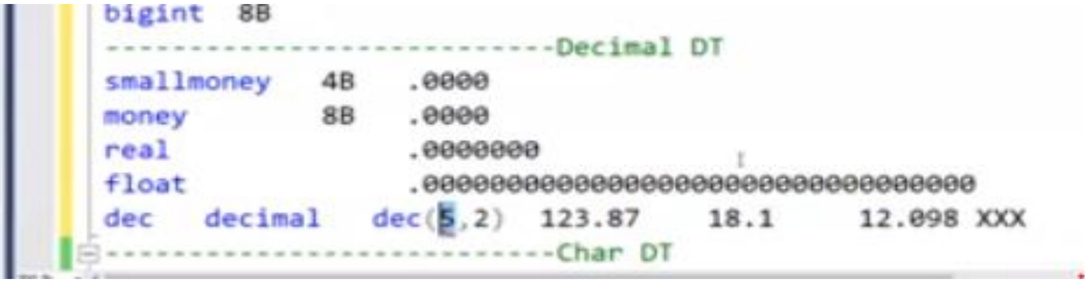
Row_Number()
Dense_rank()
NTiles(Group)
Rank()

```
Select *
From(Select *,Row_Number() over(partition by did order by esal desc) as RN
             ,Dense_Rank()  over (Partition by did order by esal desc) as DR

     From employee  ) as newtable

Where RN=1                   DR=1

      RN=3                   DR<=2
```

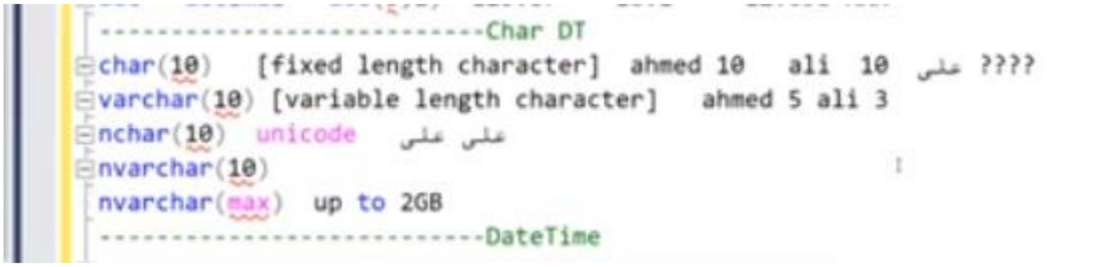
eid	ename	esal	did	RN	DR
15	ahmed	10000	10	1	1
14	ali	10000	10	2	1
12	eman	9000	10	3	2
1	nada	9000	10	4	2
2	reem	9000	10	5	2
3	khalid	8000	10	6	3
7	mohamed	17000	20	1	1
8	sayed	17000	20	2	1
6	hassan	6000	20	3	2
5	omar	6000	20	4	2
9	sally	15000	30	1	1
10	shimaa	4000	30	2	2
11	hana	4000	30	3	2
12	lama	3000	30	4	3



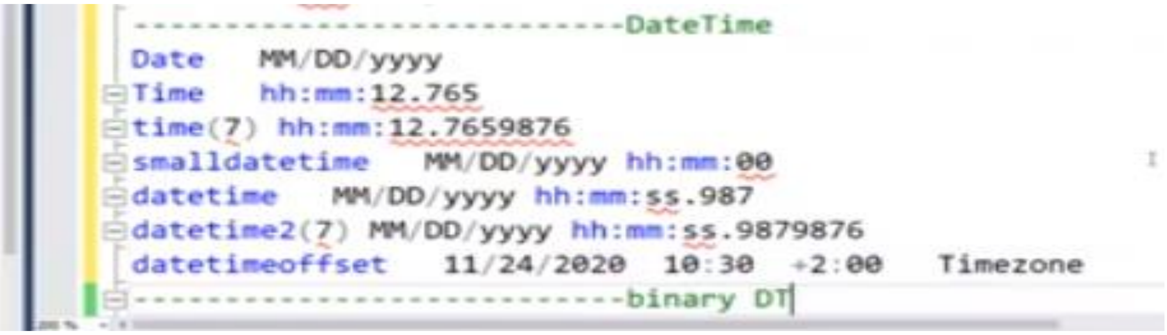
```
------------------------------------Data types
--------------------------Numeric DT
bit      --bool     0:1     true:false
tinyint 1 Byte     -128:+127    unsigned    0:255
smallint 2B        -32768:+32767  unsigned    0:65555
int      4B
bigint   8B
--------------------------Decimal DT
```



```
bigint   8B
--------------------------Decimal DT
smallmoney    4B    .0000
money         8B    .0000
real                .0000000
float               .000000000000000000000000000000000
dec    decimal    dec(5,2)  123.87     18.1      12.098 XXX
--------------------------Char DT
```

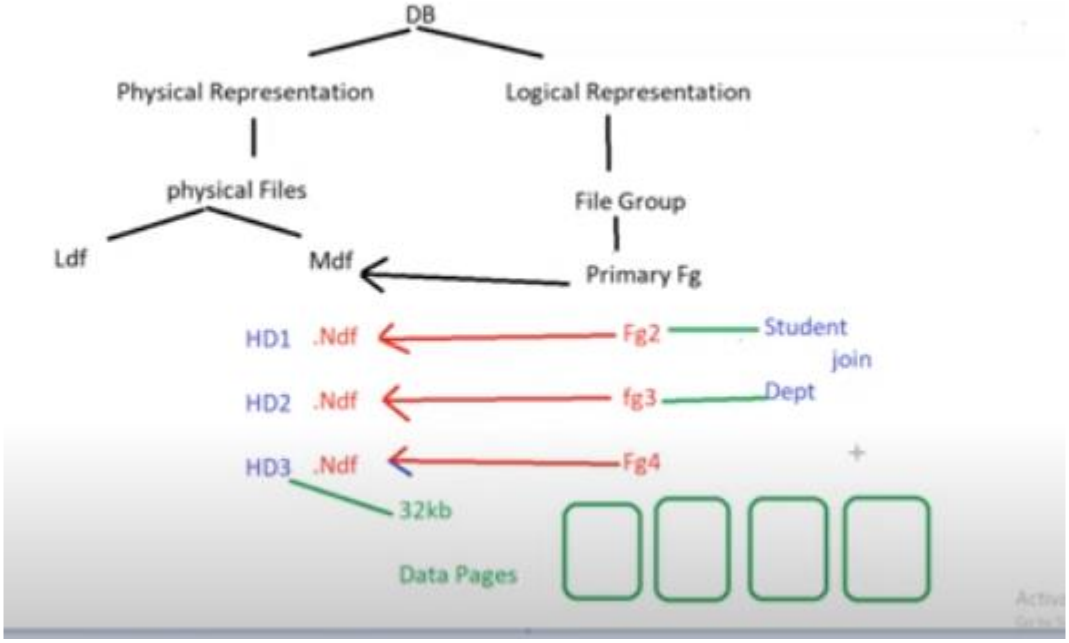


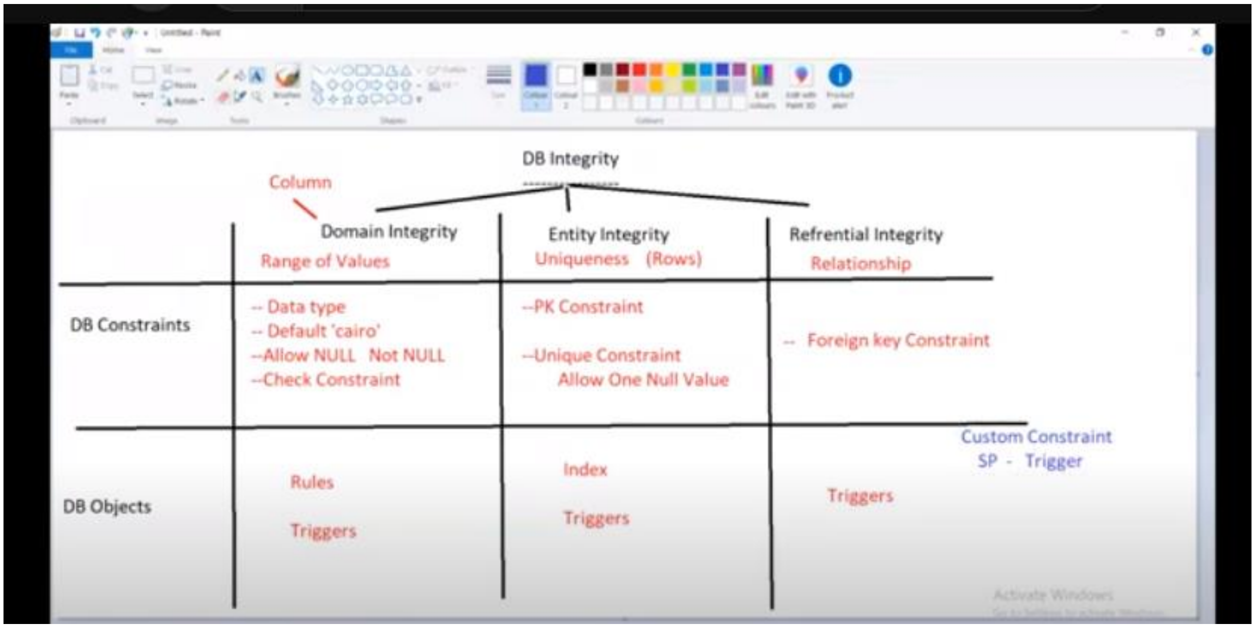
```
--------------------------Char DT
char(10)    [fixed length character]  ahmed 10   ali  10  على ????
varchar(10) [variable length character]   ahmed 5 ali 3
nchar(10)  unicode    على على
nvarchar(10)
nvarchar(max)  up to 2GB
--------------------------DateTime
```



```
--------------------------DateTime
Date     MM/DD/yyyy
Time     hh:mm:12.765
time(7)  hh:mm:12.7659876
smalldatetime    MM/DD/yyyy hh:mm:00
datetime    MM/DD/yyyy hh:mm:ss.987
datetime2(7)  MM/DD/yyyy hh:mm:ss.9879876
datetimeoffset    11/24/2020  10:30  +2:00    Timezone
--------------------------binary DT
```

6-SQL, Database Constraints, Rules-- Create DB





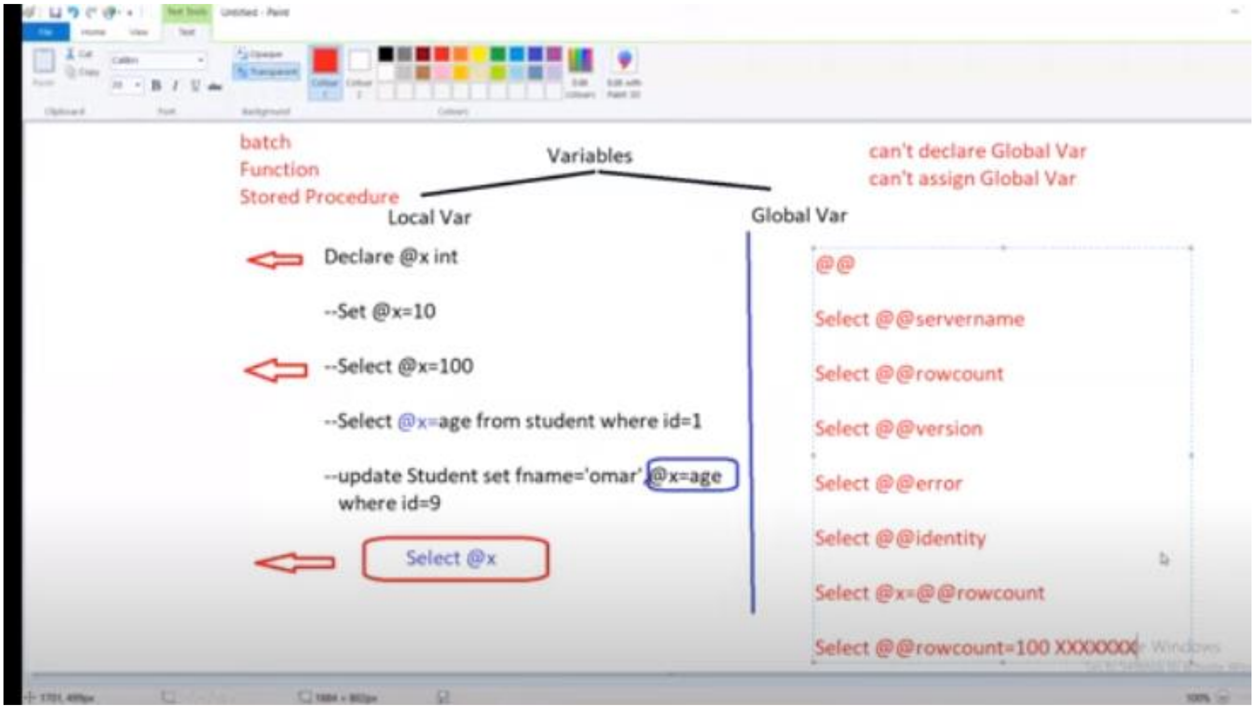
DB Integrity

Column

	Domain Integrity Range of Values	Entity Integrity Uniqueness (Rows)	Refrential Integrity Relationship
DB Constraints	-- Data type -- Default 'cairo' --Allow NULL Not NULL --Check Constraint	--PK Constraint --Unique Constraint Allow One Null Value	-- Foreign key Constraint
DB Objects	Rules Triggers	Index Triggers	Triggers

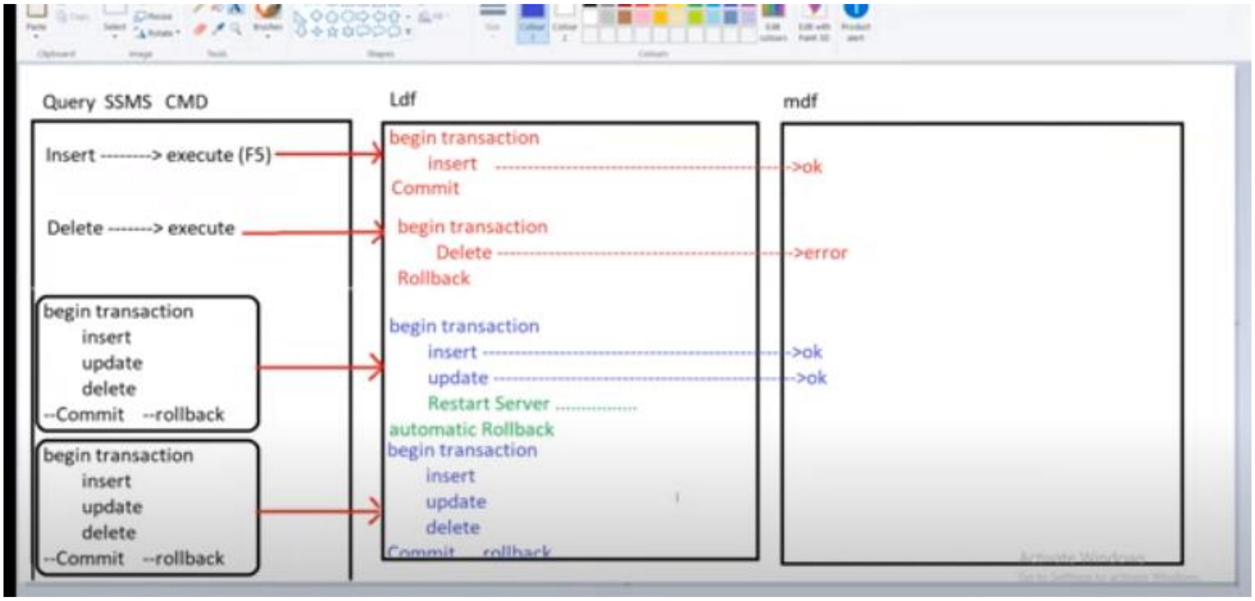
Custom Constraint
SP - Trigger

7-SQL, Variables, If, While, functions



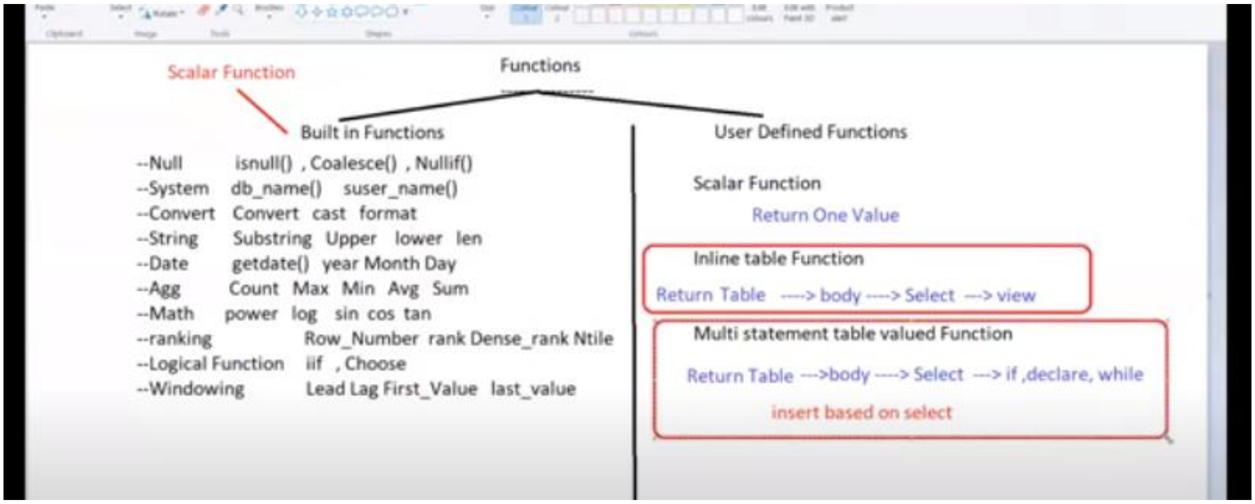
```
Declare @x int

--Set @x=10

--Select @x=100

--Select @x=age from student where id=1

--update Student set fname='omar',@x=age
  where id=9

Select @x
```

```
@@

Select @@servername

Select @@rowcount

Select @@version

Select @@error

Select @@identity

Select @x=@@rowcount

Select @@rowcount=100 XXXXXXX
```



```
Insert --------> execute (F5)

Delete -------> execute

begin transaction
      insert
      update
      delete
--Commit   --rollback

begin transaction
      insert
      update
      delete
--Commit   --rollback
```

```
begin transaction
      insert  ------------------------------------->ok
Commit

begin transaction
      Delete ------------------------------------->error
Rollback

begin transaction
      insert ------------------------------------->ok
      update ------------------------------------->ok
      Restart Server .................
automatic Rollback
begin transaction
      insert
      update
      delete
Commit     rollback
```



Functions

Scalar Function

Built in Functions

--Null isnull() , Coalesce() , Nullif()
--System db_name() suser_name()
--Convert Convert cast format
--String Substring Upper lower len
--Date getdate() year Month Day
--Agg Count Max Min Avg Sum
--Math power log sin cos tan
--ranking Row_Number rank Dense_rank Ntile
--Logical Function iif , Choose
--Windowing Lead Lag First_Value last_value

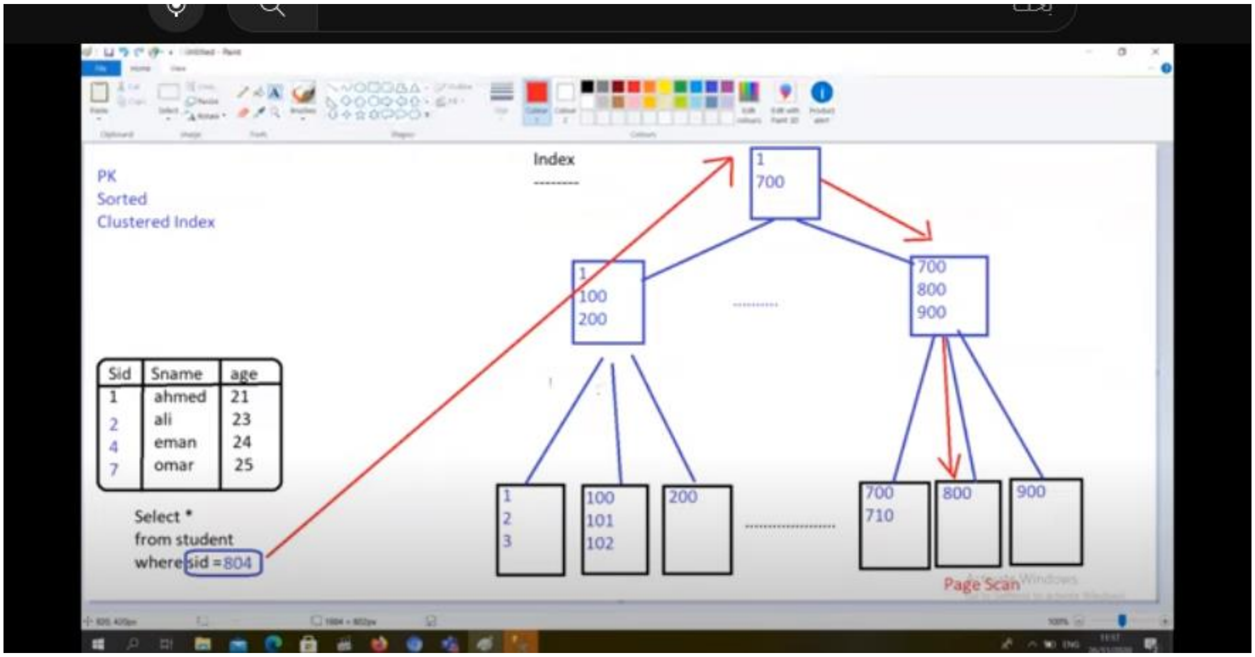
User Defined Functions

Scalar Function
Return One Value

Inline table Function
Return Table ----> body ----> Select ---> view

Multi statement table valued Function
Return Table --->body ----> Select ---> if ,declare, while
insert based on select

8-SQL, View, Index, Merge Statements, Pivot tables.

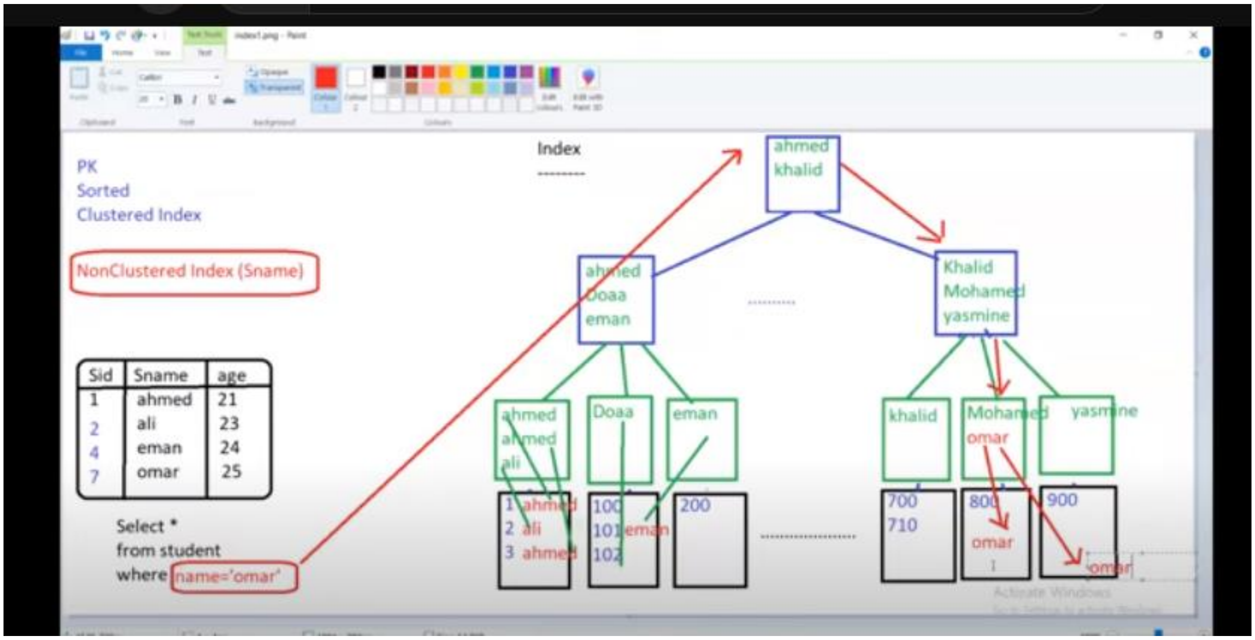


Index

PK
Sorted
Clustered Index

Sid	Sname	age
1	ahmed	21
2	ali	23
4	eman	24
7	omar	25

Select *
from student
where sid =804



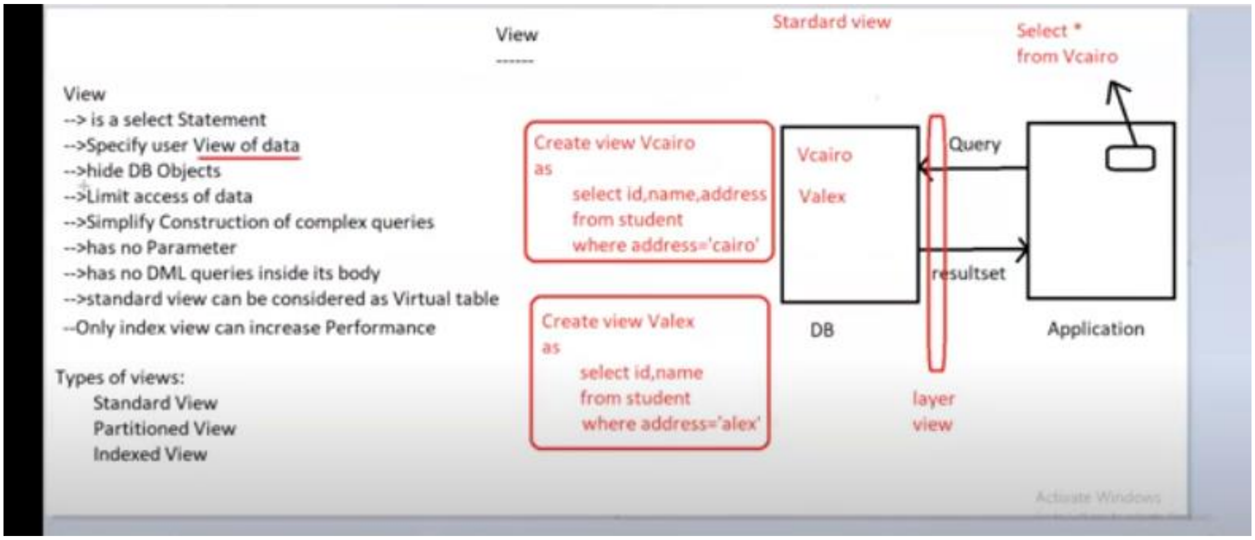
PK
Sorted
Clustered Index

NonClustered Index (Sname)

Sid	Sname	age
1	ahmed	21
2	ali	23
4	eman	24
7	omar	25

```
Select *
from student
where name='omar'
```

Index



View

View
--> is a select Statement
-->Specify user View of data
-->hide DB Objects
-->Limit access of data
-->Simplify Construction of complex queries
-->has no Parameter
-->has no DML queries inside its body
-->standard view can be considered as Virtual table
--Only index view can increase Performance

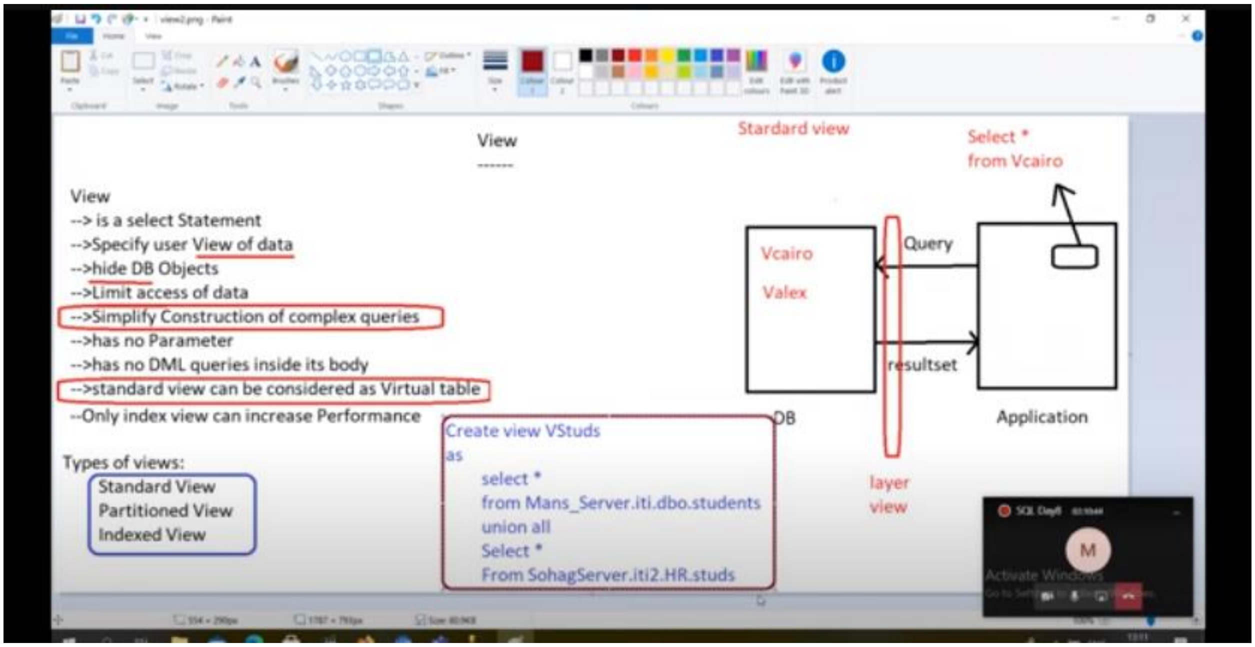
Types of views:
Standard View
Partitioned View
Indexed View

Stardard view

```
Create view Vcairo
as
     select id,name,address
     from student
     where address='cairo'
```

```
Create view Valex
as
     select id,name
     from student
     where address='alex'
```

```
Select *
from Vcairo
```

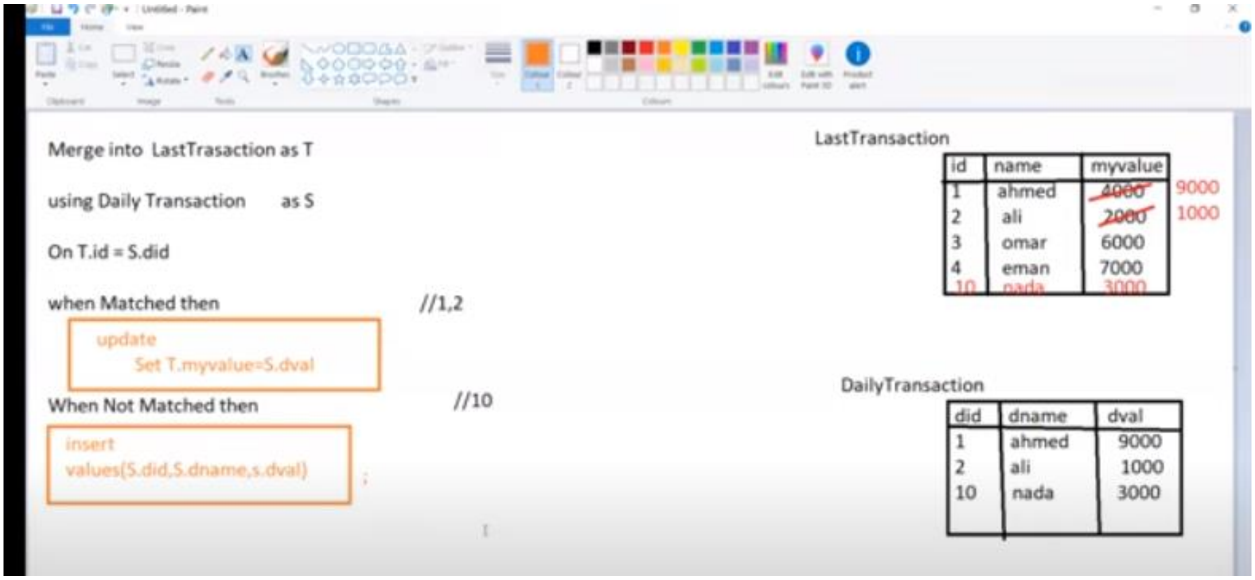


View

View
--> is a select Statement
-->Specify user View of data
-->hide DB Objects
-->Limit access of data
-->Simplify Construction of complex queries
-->has no Parameter
-->has no DML queries inside its body
-->standard view can be considered as Virtual table
--Only index view can increase Performance

Types of views:
- Standard View
- Partitioned View
- Indexed View

```
Create view VStuds
as
     select *
     from Mans_Server.iti.dbo.students
     union all
     Select *
     From SohagServer.iti2.HR.studs
```



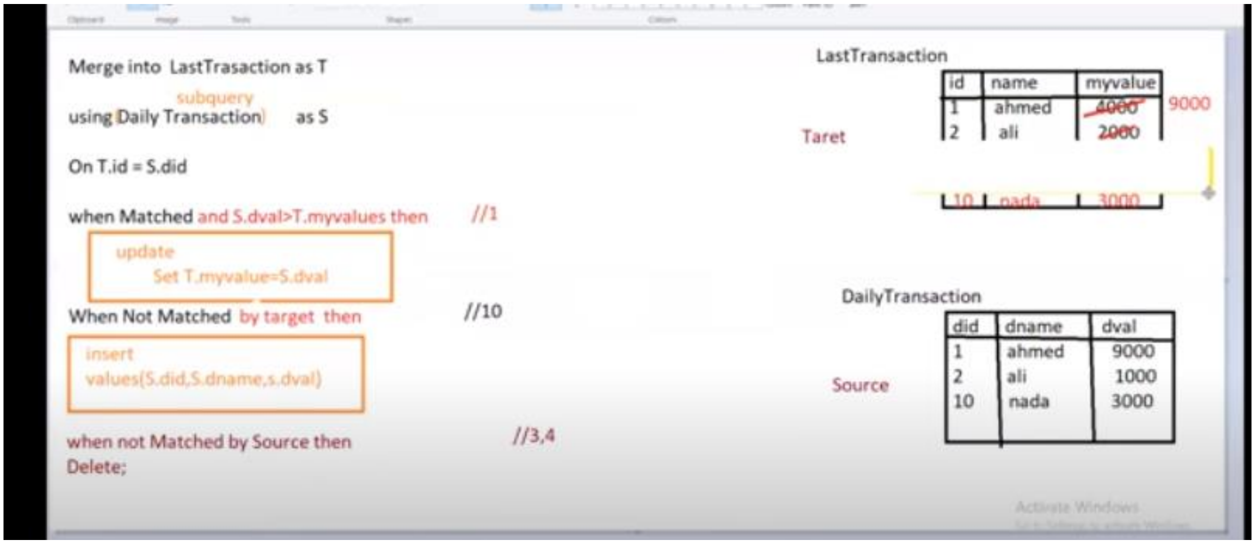
```
Merge into  LastTrasaction as T

using Daily Transaction      as S

On T.id = S.did

when Matched then                         //1,2
    update
          Set T.myvalue=S.dval
When Not Matched then                       //10
  insert
  values(S.did,S.dname,s.dval)         ;
```

LastTransaction

id	name	myvalue
1	ahmed	~~4000~~ 9000
2	ali	~~2000~~ 1000
3	omar	6000
4	eman	7000
10	nada	3000

DailyTransaction

did	dname	dval
1	ahmed	9000
2	ali	1000
10	nada	3000



```
Merge into  LastTrasaction as T
                subquery
using Daily Transaction       as S

On T.id = S.did

when Matched and S.dval>T.myvalues then          //1
    update
        Set T.myvalue=S.dval
When Not Matched  by target  then                //10
    insert
    values(S.did,S.dname,s.dval)

when not Matched by Source then                  //3,4
Delete;
```

LastTransaction

Taret

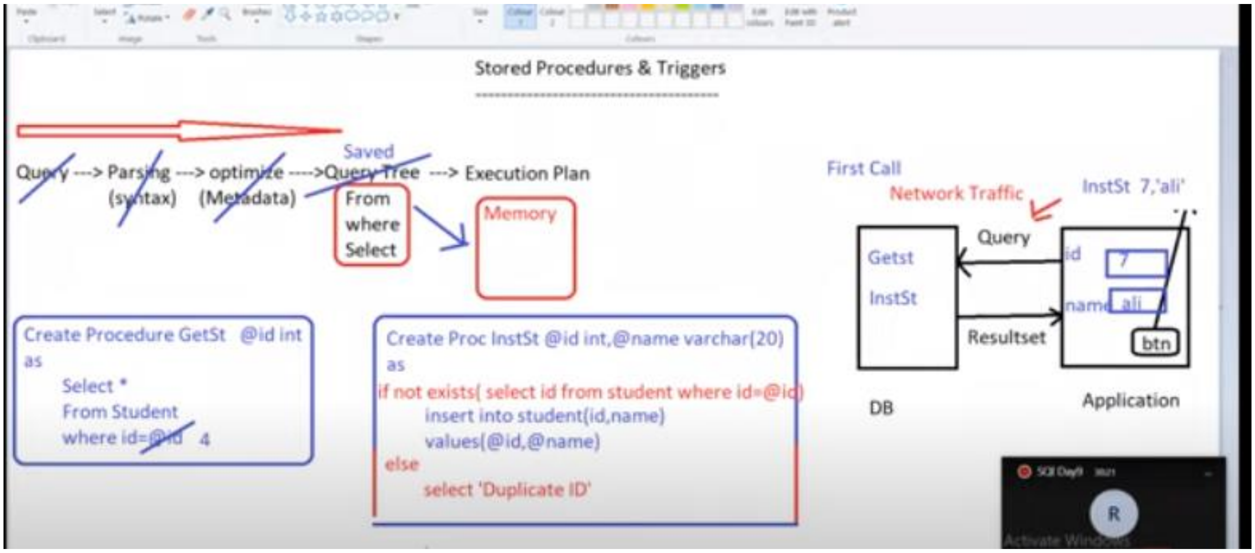
id	name	myvalue	
1	ahmed	~~4000~~	9000
2	ali	~~2000~~	
10	nada	3000	

DailyTransaction

Source

did	dname	dval
1	ahmed	9000
2	ali	1000
10	nada	3000

9-SQL, Stored Procedure, Triggers, XML Tables



Stored Procedures & Triggers

~~Query~~ ---> ~~Parsing~~ ---> ~~optimize~~ ---->~~Query Tree~~ ---> Execution Plan
~~(syntax)~~ ~~(Metadata)~~

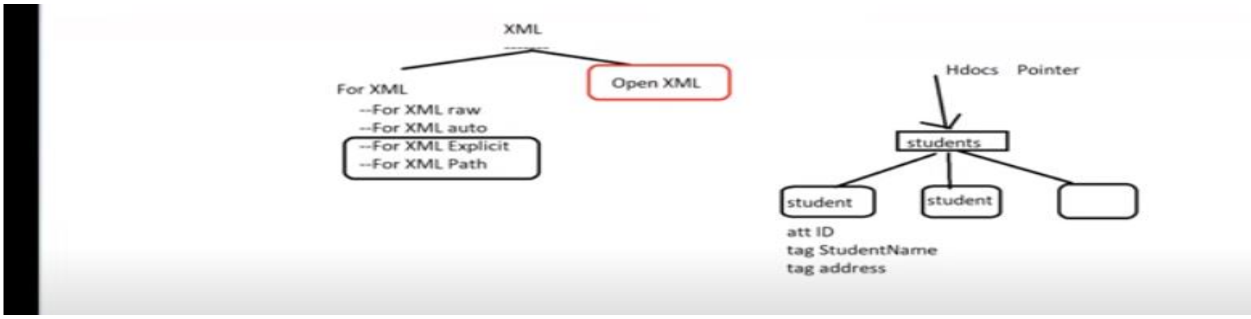
Saved

From
where
Select

Memory

```
Create Procedure GetSt   @id int
as
     Select *
     From Student
     where id=~~@id~~  4
```

```
Create Proc InstSt @id int,@name varchar(20)
as
if not exists( select id from student where id=@id)
        insert into student(id,name)
        values(@id,@name)
else
        select 'Duplicate ID'
```

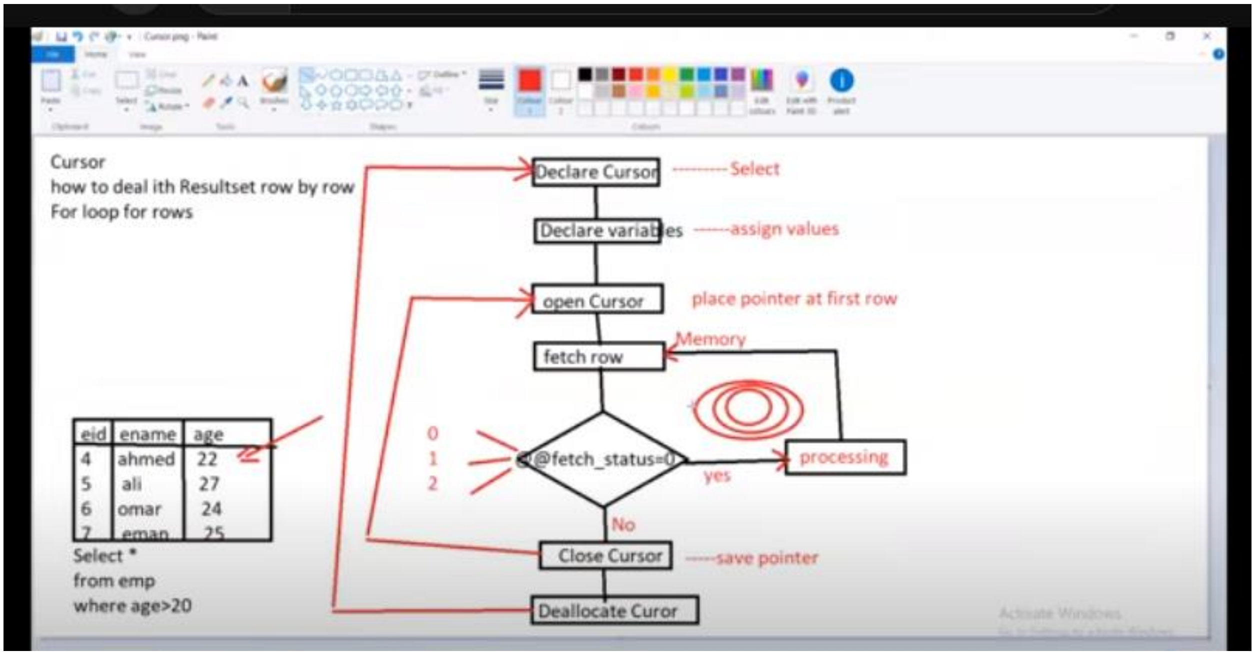


XML

For XML
--For XML raw
--For XML auto
--For XML Explicit
--For XML Path

Open XML

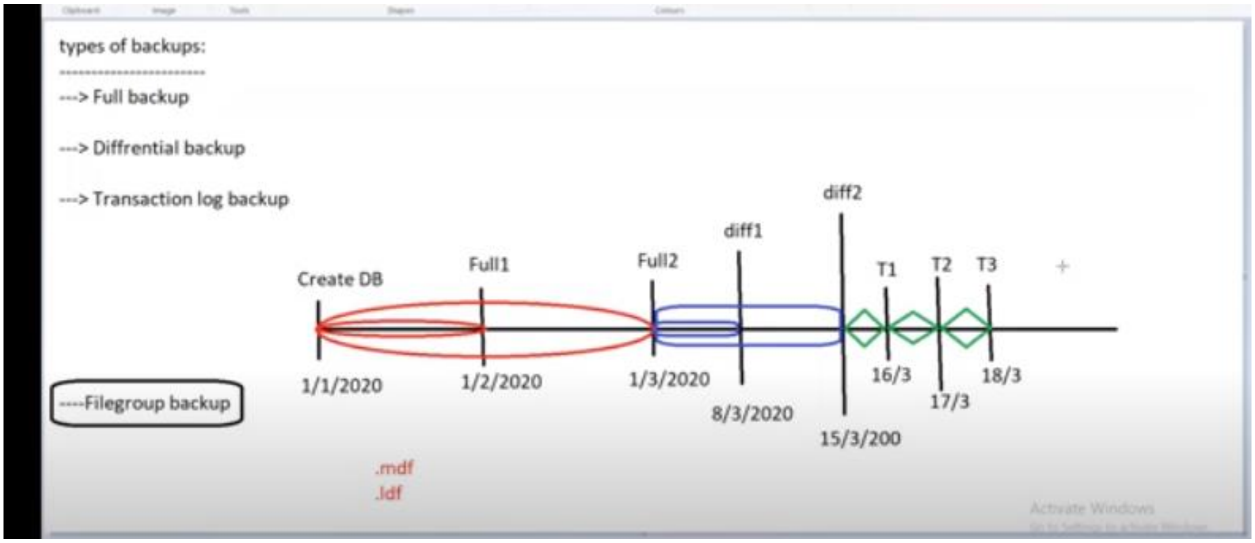
10-SQL, Cursor, Database Backups and Restore, SQL Jobs, snapshot, SQL CLR



Cursor
how to deal ith Resultset row by row
For loop for rows

eid	ename	age
4	ahmed	22
5	ali	27
6	omar	24
7	eman	25

```
Select *
from emp
where age>20
```



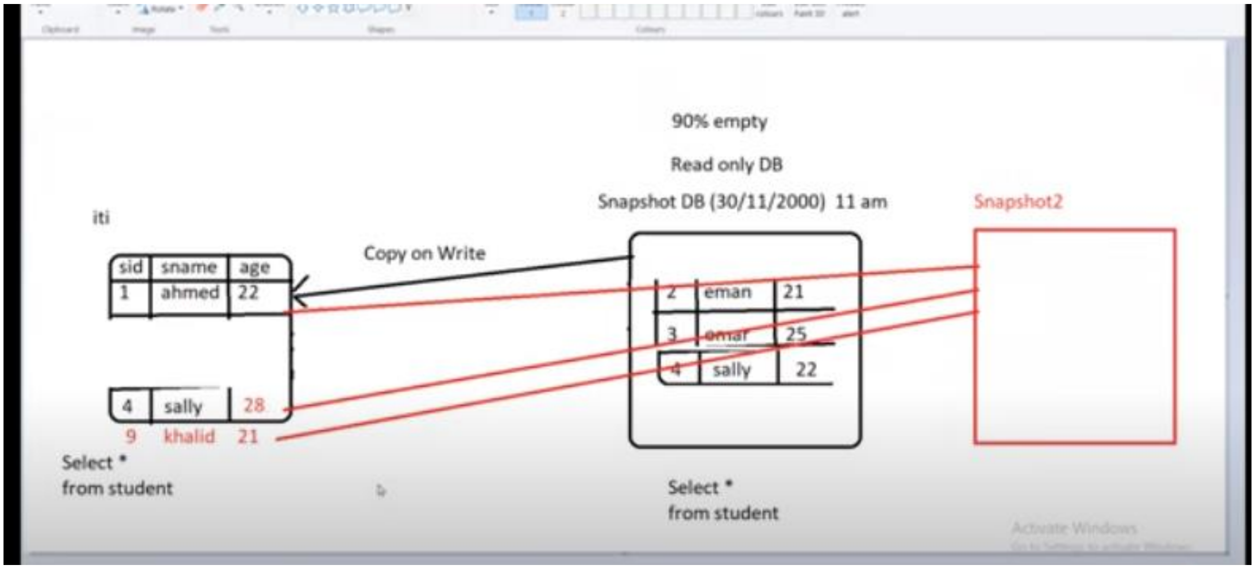
types of backups:

---> Full backup

---> Diffrential backup

---> Transaction log backup

----Filegroup backup



sid	sname	age
1	ahmed	22
4	sally	28
9	khalid	21

```
Select *
from student
```

2	eman	21
3	omar	25
4	sally	22

```
Select *
from student
```