

Day 1

⇒ 1- SQL Introduction, DB Design, DB System, ERD, File Based System ⇐

II] DB Life Cycle:

* من دزم كل الـ applications تكون بطيء في الـ background او DB شائعاً زائداً في الـ office والـ calculator مبني على DB System life cycle فيه DB هيكل لا ينتهي

① Analysis (System Analyst) : مسؤول عن المطلوب من الـ system requirements وـ the analyst

Requirement Document علثاً يعرّف الـ project scope وـ يكتب في client needs

② DB Design (DB Designer) ERD وـ يحوله لـ Req Doc (DB Designer) Entity Relationship Diagram

→ اد هناد هو الاول basic objects في System روى System Entity

مستشفى صوال all entities موجودة هي الـ Doctor وـ nurses وـ

③ DB Mapping (DB Designer) ERD → set of Rules DB Designer (DB Designer) وـ هيطبع منه الـ Tables II Physical schema II Actual ERD من Req Document الـ mapping يطبع منه الـ schema واحدة

④ DB Implementation (DB Developer) Physical Table (DB Developer) قدر تتعامل معه physical table تحتاج Tool المهمي الـ RDBMS نرمي (Relational DB Management System) SQL (structure) SQL mySQL وـ Oracle SQL Server وـ الـ DB developer يعمل كـ DB Shared DB لـ DB تكون

⑤ Application App program interface GUI او الـ Application وـ يفتح him الـ User يتابع الـ DB Application Server

⑥ Client (End user) URL يتعامل مع الـ APP في الـ browser connect to App server

قبل الـ DB كانوا يستخدمون File Based System Fixed Width File او

عيوب الموضوع دا ان الـ search في المـ data جو الـ files دا صعب طلب وـ محدود الـ performance

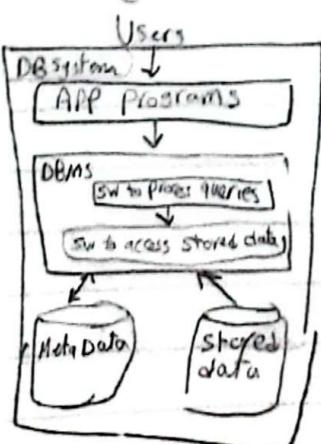
Shared file كل ما حصل ينزل على هـ file الباض مبني هيستوفـه على طول DB Integrity مبني بين الـ files وـ relationship

← ينبع من تكرار أو Duplication وـ DB هي أداة Security
 ← معيين Constraints أو Rules يعني لو استطلاع متغير يكون بين 20-30 من
 ← معيين Data Quality يعني جتن يتكت معيين string, integer وـ text
 ← لذ ينبع في كل حاجة اربط Backup ولا Restore
 ← No standard لشكل الـ files قد txt, Excel ...
 ← Diff integration وهذا يعني DB system ←
 ← بعد كذا ظهر DB system ←

← وـ Relationships بين Tables ←
 ← ميزات الموصق دا: ① يشغل على one standard ←
 ← Data about Data هو ال MetaData ← ② عند الـ DB نفسها ، والـ MetaData هي
 ← Relations, columns, Tables ← data description ←
 ← كل دى Table هو Table ← values ←
 ← Metadata كـ data غيره طريق data ←

← كل Column لازم يكون له datatype ← ③
 ← كل column يكون Primary key في Table ← ④
 ← ما يغتصب يكون قيمة غير معرفة (Duplication Redundancy) ←
 ← وفي Foreign key ← *
 ← Centralized DB ← ⑤

← الفرق بين DB system, Database Management System وـ DataBase ←
 ← Collection of related data هو DB ← ①
 ← oracle وـ SQL Server الـ DB الـ Lguage الـ Tool ← ②
 ← DB الـ DBMS Software الـ DBMS هو DB system ← ③



: DBMS Advantages ⇐

SQL بمتطلباتها مطبقة الـ Standardized DB *

Sharing data في جميع الأجهزة one server *

data و constraints في المدى *

Data Quality تحسين نوعية البيانات وcolumns الـ Datatypes *

data shared من دون تناقض Inconsistency *

automatic backup and restore، security *

data و Redundancy مفيدة *

PK في table كل *

كل الـ DB engine و Relations و Datatypes و Constrains موجودة جوا الـ DB و بذلك *

جم الكود الـ application هو كثيف و غير ممتع للـ developer

نوع الـ DB

: DBMS Disadvantages ⇐

الـ DB نفسها غالباً *

+ لازم يكون الـ developer مختركة في التعامل مع الـ DB

+ الـ DBMS و incompatible مع بعضها البعض

+ الـ DBMS و incompatible مع بعضها البعض (معظمها MySQL و Oracle و SQL Server)

Entity Relationship Diagram (ERD) ⇐

* بطاقة عن طريقة تحويلها إلى ERD لوصف العلاقات components, Requirements Document

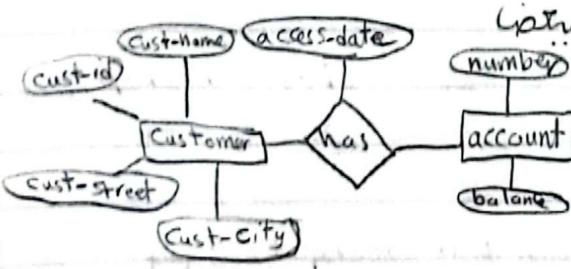
بيانها

الـ ERD يتكون من 3 حاجات:

Entities : داعي، و هي أي System أو Component في الواقع محتوى آخر بعد مدخلات

Attributes : أي خصائص لها entity مثل name و id أو cust_id و age

Relationships : صفات الـ entities



Relationships : علاقات تربط entities

: ERD نتائج الـ

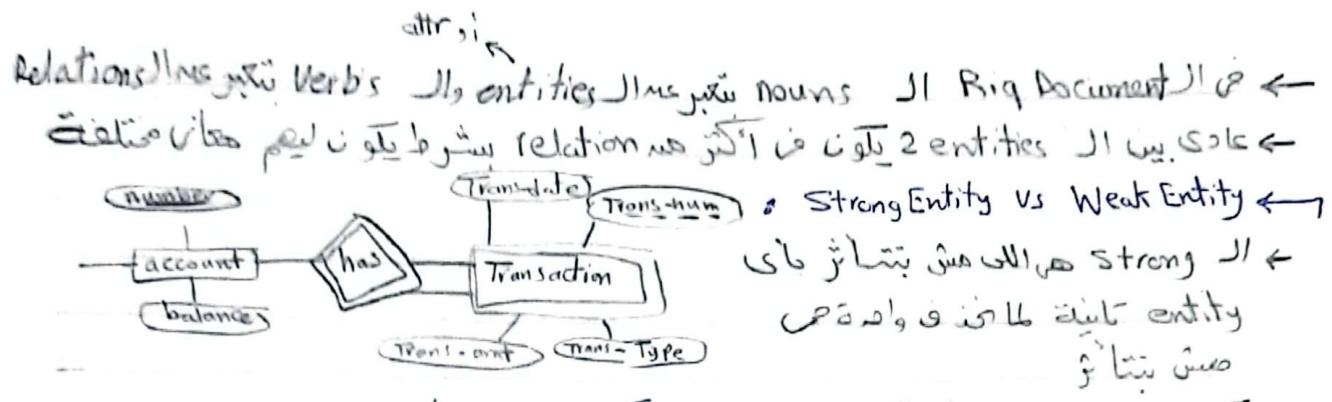
المستطيل : هو الـ Entity

المربع : الـ Relationship

الـ مصري : الـ attributes

الـ shared Relation موجود حقوق الـ attribute like access-date

بين اثنين entities و مصري بين مجموعتين entities



للدار weak entity هي التي تكون بـ parent entity

منها اد ERD دا لو هي بنك . هتخذى account عيل معنی هيتحتاج فمعاه كل المعاشر account weak entity دا ادعى نكا account تـ Transaction

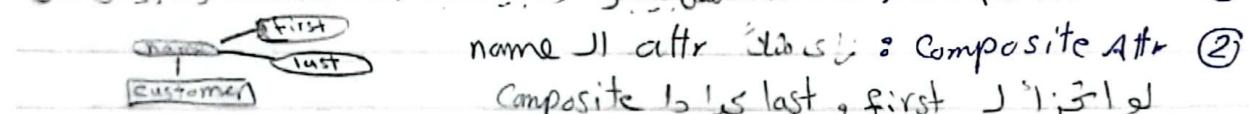
علشان دا بـ partial key جوا اتنى مستطيل والـ relation بين اتنى معنی

partial key weak or weak P.K or primary key لها strong Entity *

strong entity بـ P.K Partial Key

: Types of Attributes

ولا يمكـن لـ Simple Attr ①



name attr يـ Composite Attr ②

لـ first, last, age

olden runtime attr Derived Attr ③

net salary, date of birth, name's age

dashed lines salary date of birth name age net salary date of birth name age

بـ Multi-valued Attr ④

الـ customer زـ اي رقم المـ phone-num والـ address city country

واحد يكون له اكتـ معـ phone-num address city country

بـ Complex Attr ⑤

multi valued attr greet city country customer

address customer city country customer

* دا هو الـ Verb او الـ adjective اللي بيـ join الـ entities

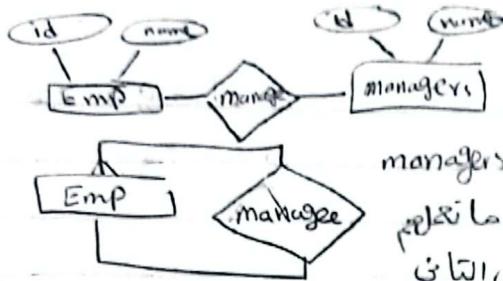
2 entities relational يـ join في المـ relationship

entitieـ Shared relationship attr join +

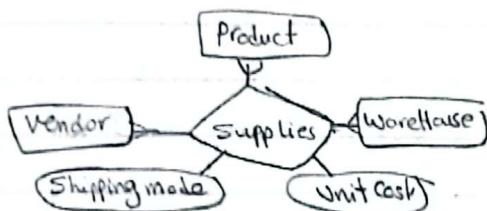
* Relationship has 3 properties

: Degree of Relation ①

(Ternary - binary - unary) 3 ways 3 types 3 degrees 3 forms 3 ways 3 forms 3 types 3 ways



الـ ① الـ Recursive Relationship of Unary
يُمثل في نفس الفرع زاد الـ managers, Emp, managers
سيتم علاقة manage ولديهم نفس الأسماء attributes مثل ما نعلم
بالشكل الثاني إلى صور يختصرهم دوري entity واحدة بالشكل الثاني



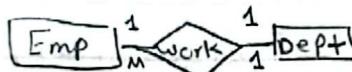
الـ ② الـ Binary Relationship
between two entities

الـ ③ Ternary Relationship

الـ ④ Cardinality relation
 بين مدخلات أكثر من

* يحصل طبقاً على متغير shared بين هذا relationship لارم خط كل دى نفس الـ entities كل entity

الـ ⑤ Cardinality

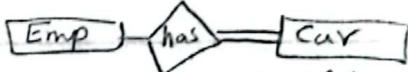


يعنى عدد الترابطين الـ entities يعني عدد
الـ depts 1، Emp دا يقول كل Dept يستعمل في واحد عددهم هذل 1 عند الـ M
وي Russo يقول كل Dept يستعمل فيه أكثر من عددهم 1 عند M
الـ Emp

وكذا الـ Dept عنده 1 وار Dept M(Many) دا Emp
عادي الـ 2 entities manages relationship
يتكون منه 3 relations 1-1

كما الحال الـ rows كل entity لها
cardinality one table

(M - M , 1 - M , 1 - 1)



الـ ⑥ Participation

يعنى كل الـ rows بتاعت الروابط ممكنها العلاقة ولابد

لوكا لهم داخلين يعني must وينبغي عطه بـ Total participation

some أو may فوالـ rows بس اللي داخل يعني Partial participate وينبغي عطه بـ

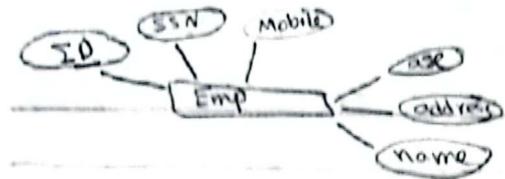
Emp may has a car على أي حال حفظوا
في المثال دا معنى دا معنى Emp may has a car

وكلـ car must be assigned to Emp

Total participation دا يعني لها 2 lines لا زالت الـ weak Entity

الـ strong Entity

الـ zero or more أو optional أو mayPartial



:keys ←

: Candidate Key ⑧

هي المرحلة التي تسبق اختيار الـ PK

↳ يعنى ذلك أن Attr كذا فيه ID، الـ SSN والـ Mobile الثلاثة حول يتحققو شرط PK لأنهم unique، Not null، PK للثلاثة ملوكه PK، Candidate keys هلاً؟! ID هو الـ Primary key ②

لومين موجود اد ID, SSN, الار, الال, Mobile ↳

* يحصل على key : Composite Key (3)
 الـ PK ، له 2 attr دوں میں ایک age و Name میں ایک دوں میں ایک address و name میں ایک
 کامپلیکس کیا کرنا کہ اس کے لئے اس کے 2 attr کو ایک دوں میں ایک دوں میں ایک
 کامپلیکس کیا کرنا کہ اس کے لئے اس کے 2 attr کو ایک دوں میں ایک دوں میں ایک

Day 2

⇒ SQL Mapping, DB schema, SPL statements, Creating DB ←

→ DB Life Cycle:

- ① Analysis: اول مرحله او وشونا در client ایمه الـ هيطلبها وایم analysis requirement document
- ② DB Design: Req Doc مطلوب، system analyst دو حواله ایکلام دا حواله requirement document تابي حاجة هيعلمها او Req Doc در، هو انه یا پس او DB Designer لـ ERD و هیچو له د
- ③ DB Mapping: ERD لـ rules برضو هيطبق شويه DB Designer ۳
- ④ DB Implementation: developer جل جل کړي، الـ implementation physical DataBase بدل او SQL ۴
- ⑤ Application: client application GUI ۵
- ⑥ Client: end user

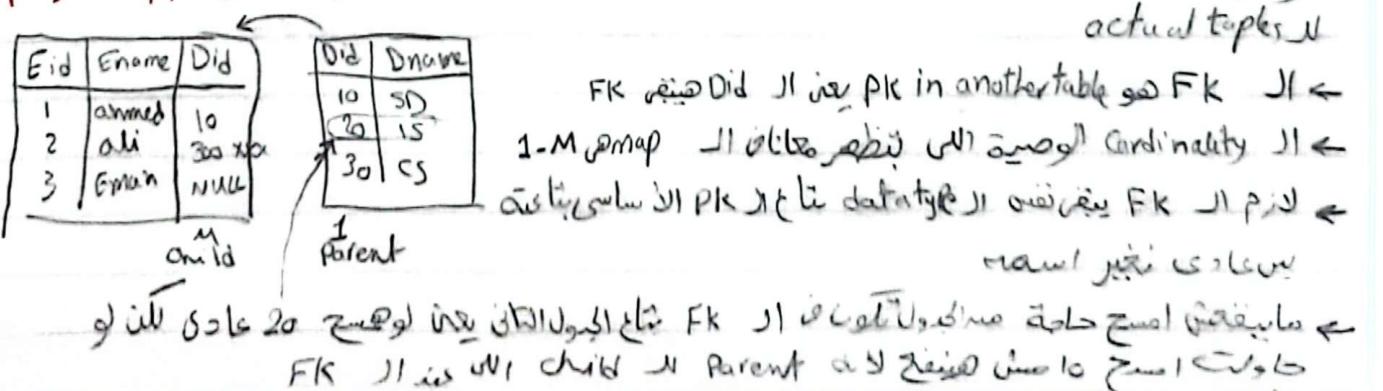
→ ERD

- ① Entity: بناء الـ data بنائها Entity دی : strong Entity ۱-۱
PK (primary key) دانتانیة علیو، لدیها هنستائز د بـ delete
- ② partial key لـ Entity ۲-۲: دی و هو دها بـ معنی دی ویکلون لهیانه ویکلون لهیانه
- ③ attributes: characteristics دی اویا columns دی اویا
multi valued ۲-۳ composite ۲-۲ Simple Attr ۲-۱
Complex ۲-۵ computed ۲-۴

③ Relationship:

- (Ternary - Binary - unary) Degree ۳-۱
- (M M - 1 M - 1 1) Cardinality ۳-۲
- (total - partial) participation ۳-۳

→ Mapping:



يبقى الـ datatype allowable in the table FK \leftarrow
 أي قيمة في الـ FK تكون تذكر أو تأخذ NULL سلازم values all or
 (FK \rightarrow PK in child table \rightarrow match PK in parent table)
 مثلاً: العلاقات في الـ DB \rightarrow ما يلي
 $\text{child} \rightarrow \text{parent} \rightarrow \text{delete}$ \leftarrow
 $\text{partial part} \rightarrow \text{allo} \rightarrow \text{NULL}$ \rightarrow FK \rightarrow
 $\text{range into range of values, and/or Domain of table in column of}$ \leftarrow
 any value

- $\text{All columns must have data type}$
 (int as integer quality يبيح جابين هم الـ datatype)
 $= 128; 127$ and $1B$ and $tinyint$ to size size \rightarrow وبيحه الـ datatype \rightarrow الـ $\text{constraints & Rules}$
 - لازم يرضي يكون في $\text{Constrains & Rules}$

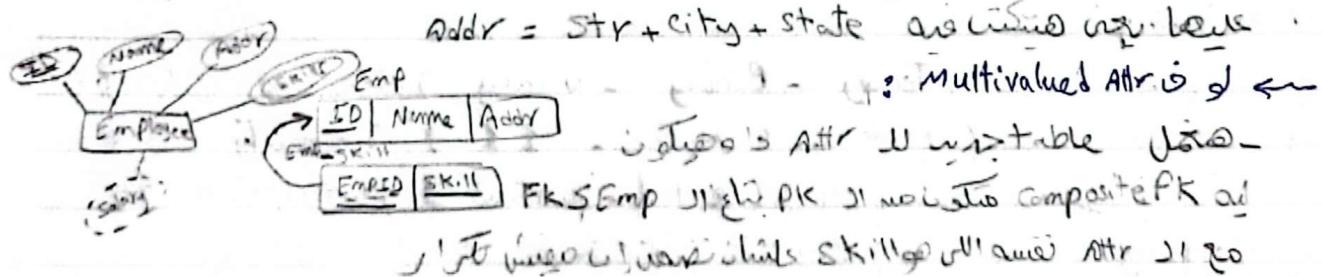
→ ER-to-Relational Mapping:

أولاً حاجة بنتوف لورقة $1-1$ \rightarrow total participate \rightarrow صن الناحيتين خالها بس
 دى منه بمنظور كتر

table NPK : اى Strong Entity \rightarrow Step1
 لو اد كامم Attr: Simple Attr \rightarrow هنزلو عادي من الجدول
 اوى Attr: Composite Attr \rightarrow الـ HINZEL هو المكونات بناءً الـ Attr دا متعددة



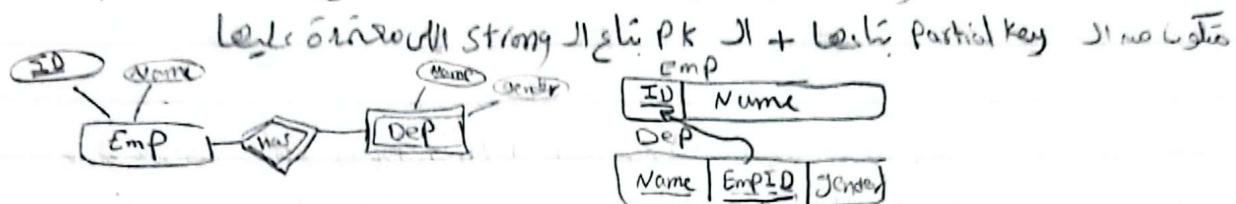
كذا Address موجود بس فتح لهم مع الـ file schema \rightarrow file schema \rightarrow Customer \rightarrow Address
 \rightarrow Multivalued Attr \rightarrow فيه كل الـ rules , constraints , equations \rightarrow Constraints file



لو من Salary , Skill \rightarrow Deriven Attr
 سلسلة كامم فتح لها table \rightarrow Storage \rightarrow table \rightarrow equation
 سلسلة كل المكونات \rightarrow table \rightarrow deriven
 المكونات \rightarrow table \rightarrow Attr \rightarrow الـ equation
 اد لو في جدول تاني يعني على الـ Attr دى

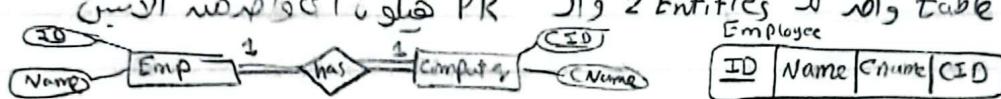
Step 2) Mapping of weak Entity :

- اول حاجة هى table له key strong ىعنى مفتاح مركب
Composite PK له key مركب يتكون من 2 أو 3 key (partial key) weak



Step 3) Mapping of Binary 1:1 Relation Types:

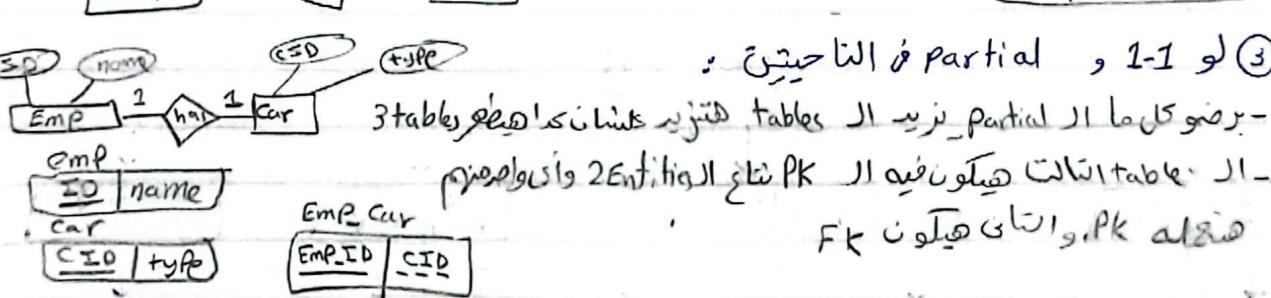
① لو كانت total participation 1-1
يجب أن يكون كل من_entities له key PK
يتكون من 2 key (partial key) وال 2 Entities ينتمي each table



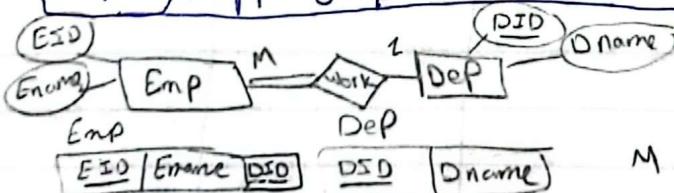
② لو كانت 1-1 ، partial 1-1 ، 1-1

هذا يعني tables كلها كلها partial key

هذا يعني كل key كلها partial key



Step 4) Mapping of 1:M Relationship



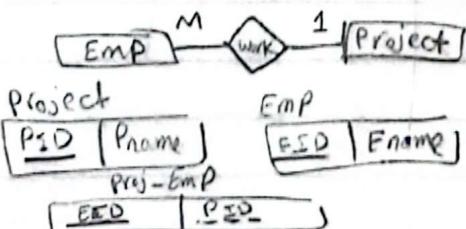
many ىعنى *
لو كانت 1:M ، 1:M

M 1:M PK also 1 key

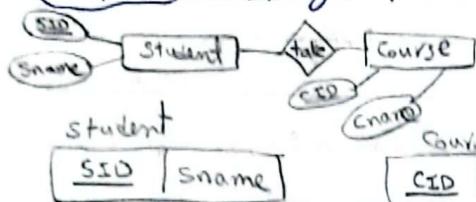
partial 1:M 1, 1:M

كل key 1:M 1:M

M 1:M PK also PKs 1:M



Step 5 Mapping of M:M Relationship Types:



مُنْهَبِيْسْ يَا ار جَوْلِ الْمُنْهَبِيْسْ، participation ار جَوْلِ الْمُنْهَبِيْسْ
PK المُنْهَبِيْسْ الْأَنْتِيْنْ composite attr PK
الْمُنْهَبِيْسْ الْأَنْتِيْنْ student-course

لَوْمَنْدِي relationship يَعْلَمْ بِنْجَلَه جَوْلِ جَدِيدِ وَلَكَنْ هَذَهْطَه فِي الْجَوْلِ
الْمُنْهَبِيْسْ هُوَ FK وَ PK سَوَادْ جَوْلِ لِيَهُمْ اُولَئِكَه FK هُوَ جَوْلِ اد

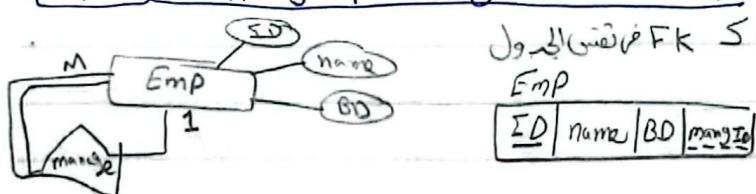
Step 6 Mapping of N-ary Relationship

Ternary Relation

participation of Cardinality
الْمُنْهَبِيْسْ جَدِيدِ فِيهِ كَلِّ الْPks وَ اَوْنَى
مُوْجَوَّهَاتِ جَدِيدِ relation مُنْهَبِيْسْ!

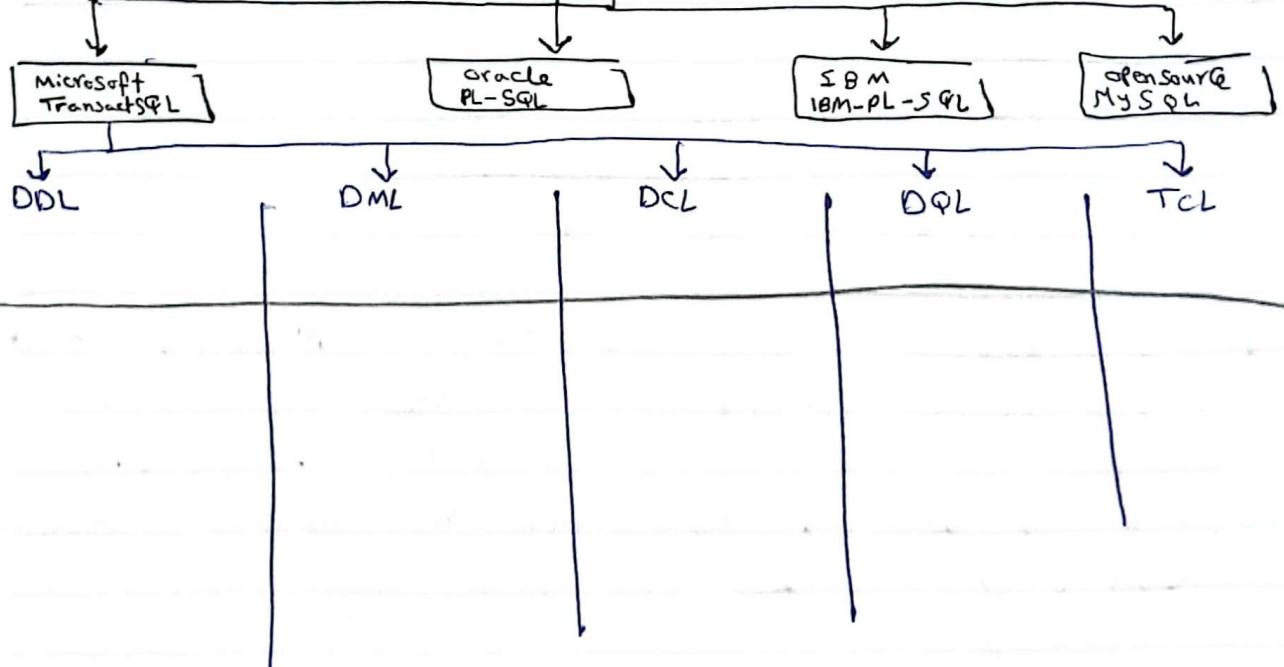
الْمُنْهَبِيْسْ مُنْهَبِيْسْ relation وَ اَوْنَى PK هَذَهْتَه وَ سَوْفَ! يَعْلَمْ اَوْنَى مُنْهَبِيْسْ

Step 7 Mapping of Unary Relationship:



خَارِجِيْسِ الْمُنْهَبِيْسْ كَلِّ FK وَ PK اَوْنَى self relation J1
relation J1 بَلِّ

ANSI SQL: 2.00



→ 2:10 : 2:22
→ 2:22 : 2:27
→ 2:27 : 2:32

wizard → SQL Server (tables) ←
diagram ← ازای نمودار ←
tables ← ازای خط ارانت اور
Composite Pk ایک table میں ایک-
new query ← ازای نویں

→ 2:32

Create DDL:

→ Create table emp

```
( eid int Primary Key,  
ename Varchar(20) not null,  
eage int,  
eadd Varchar(20) default 'Cario',  
hiredate date default getdate(),  
Dnum int )
```

get date of the system

→ add Column or edit:

→ alter table emp add Sal int

→ alter table emp alter column Sal bigint

→ delete Column from DB:

→ alter table emp drop column sal

→ delete table from DB:

→ drop table emp

لهم كون نفس نوع
البيانات
معن ملخص
char(20) Varchar
int

DML (Insert, update, delete)

→ Insert:

→ insert into emp

values (1, 'ali', NULL, 'alex', '1/1/2010', NULL)

* Insert Constructors:

→ لوضى خايرین الترتيب هنائب فرمادا ایک table میں ایک row ایجاد کرنے کا ایک طریقہ
values ('eman', 9), ('ali', 12), ('nada', 7)

→ Update:

→ update emp

set ename = 'omar', eage = 22, where eid = 1

set ename = 'omar', eage = 22

where eid = 1

→ update emp

set age + = 1

→ update emp

set ename=NULL

→ delete:

→ delete from emp

* كما نسخ كل الساٌل لـ الجدول من الجدول الذي موجود
* الآن قم بـ نسخ الجدول نفسه من الـ DB

→ delete from emp

where eid=1

→ DQL

columns لـ جدول

→ select:

all

→ select * from student

* يعرض كل الأعمدة

→ select st_id, st_fname from student : 2columns

→ select st_id, st_name from student : selects 2column

where st_age >= 25

where → rows لـ جدول

لو أتيت بأعمدة الـ ست انتهي مختلف

default all descending

ascending

→ order :

→ select * from student

order by st_age desc

⇒ Concatenate 2 columns and alias:

→ select st_fname + ' ' + st_lname as [fullname] space بين الكلمتين

↓ لـ حينا جدول student من select [fullname] from student

↓ run time في كل الأعمدة التي هي table

⇒ is not , and :

→ select * from student

where st_fname is not NULL and st_lname is not NULL

⇒ distinct :

↓ جدول واحد column II order by ↓

→ select distinct st_fname from student

⇒ OR :

→ select * from student where st_addr='cairo' or st_addr='alex'

⇒ in :

↓ first one is OR else OR

→ select * from student

where st_addr in ('cairo', 'alex', 'mansouria')

⇒ between :

→ select * from student where st_age between 23 and 25

↓ = where st_age >= 23 and st_age <= 25

3- SQL Joins, Normalization

joins

⇒ Types of Joins:

- ① Cross join (Cartesian product)
 - ② Inner join (Equi join)
 - ③ Outer join (Left outer join, Right outer join, Full outer join)
 - ④ Self join (Unary relationship)

→ ① cross join

→ Select Sname , Dname
from Student , Dept

→ خی مالکرو سوچت بتکت کدا

from Student cross join Dept

student			Deft	
Sid	Sname	Did	Did	Dname
1	ahmed	10	10	SD
2	Ikhaled	10	20	HR
3	Eman	20	30	IS
3	omar	NULL	40	Admin

- * كأمثلة على Cartesian product 2 tables يعني هيأخذ الاربع اسماً ويعتبرهم فار LDJ بعدين الاربعة معوار LR وهذا يعطون الناتج عبارتين 16 Row
- * اد testers هم اللي يستخدموا Ad Query اللي في SQL query هم cross join
- *المعروف ان Ad يطلع هم اسماء الطبيعة اللي بينهم اقسام بسيطة يعني يطلع 3 اسماً بسيط والقسم تابعهم كل تابع الرابع عليه NULL فـ Ad

→ ② Inner Join (Equi join)

my Select Sname , Dname
from Student , Dept
where Dept.did = Student.did

Sname	Dname
ahmed	SD
Ishakel	SD
Eman	HR

- $\Sigma M = 0$ حول المثلث
- $F_k = P_k$
- موجود ليها (قسام) 3 rows بين الاسناد الى كذا
- $\Sigma F_x = 0$ على المثلث

وَال PK هُوَ مُفْسِدُ الـ FK إِلَى كُلِّ طَلَبَاتِ studenٌّ تُصْنَعُ tablesٌ نَّقْرِئُهُمْ alias name تُكْرِئُهُمْ

my select Sname, Dname

from, Student S , Dept D

where J.

where $D \cdot \text{did} = S \cdot \text{did}$

نَكْتَنْجَانَ (nangtan), وَهِيَ قُصْدَقَةٌ

→ select Sname, Dname

from Student s inner join Dept D

on D.did = S.did

→ ③ Outer join

→ left outer join ← لو عايز أعرض أسماء الطالبة كلها حتى اللي مش موجودة في قسم
↳ Select Sname, Dname
from Student left outer join Dept D

* كذا عايزه اد هايل Student table هو اللى يترتب على كل الأقسام
* علشان كاسين الـ left outer join كدا

Sname	Dname
Ahmed	SD
Khaled	SD
Eman	HR
Omar	NULL

left outer join

→ Right outer join

↳ Select Sname, Dname

from Student right outer join Dept D

on D.did = S.did

Sname	Dname
Ahmed	SD
Khaled	SD
Eman	HR
NULL	IS
NULL	Admin

right outer

→ Full outer join

↳ Select Sname, Dname

from Student full outer join Dept D

on D.did = S.did

Sname	Dname
Ahmed	SD
Khaled	SD
Eman	HR
NULL	IS
NULL	Admin

* يعرض كل سما، اللي لها نسخ
وغير كذا الأسماء اللي لها نفس
قسم والذئام اللي فيه
طبع وضو

→ ④ Self join (unary relationship) self relationship table

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

from Employee X, Employee Y

where Y.eid = X.superid

Eid	Ename	Superid
1	Ahmed	NULL
2	Omar	1
3	Eman	1
4	Nada	2

* لو عايز أعرف اسم الموظف واسم المشرف على 2 columns supervisor

column alias name table Employee الـ Ename نوعهم

* ناتج الـ query دي صنفت الأسماء اللي لها Superid

* المفترض هيتحل سلة تابع من نفس الجدول دائم الصيغوري هيتحت الأسماء

X.eid child Parent Y.eid child

والثاني Y.eid child

لدي هيتحته جدول الـ Supervisors وكذا يحتوي Superid دي FK من الجدول

اللى افترضناه جديده من الصيغوري كانت اساسياً يحتوي على

Y.eid = X.superid هيتحل PK=FK

ار Self join يفتح بعدها معاها outer و inner join

☞ Select x. St_Fname as Sname , y.*)
from student x inner join student y
on y. St_Id = x. St_Super

* كا هب حرف اول Column فيه

يعد كرا هب حرف كل الاحوال
ستاع والي هو سخنة من الجدول الاصلي
بس هي عرض اد Rows الباقيون عندها
St_Id = St_Super

join Multi tables:

لهملا في direct relation بين 2 tables
وبي كل واحد فيهم كذا قبل join all
* كما تحتاج شرط join عشان اغل join بين ادل ونهاي table
تاني بين الثاني والثالث

☞ Select st_fname , crs_name , grade
from Student S , Stud_Course SC , Course C
where S.St_Id = SC.St_Id and
C.crs_Id = SC.crs_Id

* صغرى بين ال FK والPK
في كل حالة ما ستفو ال diagram
وستفو ال properties
ال relation بين كل اتنين

☞ Select st_fname , crs_name , grade
from Student S inner join Stud_Course SC
on S.St_Id = SC.St_Id
inner join Course C
on C.crs_Id = SC.crs_Id

لو هستخدمو اول 2tables inner join
وتحتير الناتج دا table جديد وهو
مع الثالث وطبعاً join
طبعاً قادر على كذا

Join DML:

لعايز اعمل join مع delete او update او insert

join , update:

☞ update Stud_Course
set grade+=10

* كا هب join للاسسا sc و s على نفس ال Id و بي كا ستفو ال Id في Cairo

from Student S , Stud_Course SC

edit grade لـ update

where S.St_Id = SC.St_Id and St_Address = 'Cairo'

join , delete , join , insert

is null) IS NULL replace it by a character like NULL → replace it by a character like 'no name' ←
→ Select isnull(st_name, 'no name') as newname
from student

→ select isnull(st_fname, st_lname) from student

* بس کہ ا لوگات میںی فہم فارل st_name گھریو NULL نہیں فارل st_name
Coalesce (leadfunc position multiple replacement) ←
وہی func تھا کہ arram 8-یں column پر فارل st_name کا

Ans Select 'StudName = ' + st_fname + ' & age = ' + Convert(varchar(2), St-age)
from Student

مثلاً في SQL يمكن إدخال الاسم كـ `John Doe` في `first_name` و `last_name` كـ `NULL`، مما يعادل إدخال الاسم كـ `John NULL Doe` في `full_name`.

→ Select (Isnull(st_fname, '') + ' ' + Convert(varchar(2), Isnull(stage, 0)))
→ This func is good for performance (أفضل في الأداء)

→ select concat(st_fname, ' ', st_age) as student_name
from student

→ like: like دی او عا, فاجز، صوالکامہ سنجکھا like ←
zero or more like, 1 char like ()

→ select * from student
where st.fname like 'ا%' وينتهاي اي حرف اى وتساءل

لـ like 'a%' *
 كـ 'a%' *
 دـ 'a%' *
 a or h or m 'ahm%' *
 m 'ahm%' *
 range > 'a-h%' *
 h range

⇒ order by

⇒ Select st_fname, dept_id, st_age
from student
order by st_address

Column by order لعمد اول
عمر الى خ الـ select
Select st_fname, dept_id, st_age
from student
order by 1
Select st_fname, dept_id, st_age
from student
order by dept_id asc, st_age desc
* كل ترتيب 1 او 2 او 3 لا يهم بحسب ما في الـ select
* دام هنا هيرتبهم الـ order تصاعدي (asc)
* جلس اول id
* بعد كذا يسوق اللي ليهم id هررررا و dept_id
* رأى بعض هيرتبهم تنازلي (desc) بحسب id
* لو اول حاجة بترتبي (تساوي) كانت id ودي حاجة افلات من بتكرر يعني
الـ Query ملهاش دارفة لا ان هيرتب على نفس id st_id بس

⇒ Normalization ↵

DB Design نعمل في طرقين :

ERD ①

Normalization ②

لو كان في سистем موجود بس فيه مسائل زي إن البيانات فيها تكرار هنعمل
normalization على الـ tables اللي موجودة من هنعمل منه الـ ERD والـ mapping
ولكن هنعمل عليهم علشان نطلع mapping مخطوط
علشان كذا مناسبة لما نعمل ERD ولما الـ mapping اللي وصل من نطبق
عليه قواعد الـ normalization كذا كي نعيش حاجة كذا
يعنى الـ normalization هي مجموعة من الـ processes اللي بتطبقها على الـ tables
هي المترافق علشان نعمل الـ duplication و الـ inconsistencies
* الـ algorithm اي هيفقسم الـ table الكبير الموجودة الى الـ subtables
* الـ relation بينهم

* والـ 4 steps هو جو عماد steps كل خطوة بتطبق rule واحدة

* اولاً بيعرف لو هي مسائل
* بيتشت في الـ duplication او الـ update او الـ insert او الـ delete
* الـ behaviour اللي المفروض في العمل فيه

* ثانية في الـ DML anomalies يعني خروج من المكتتب

=> functional dependency: $A \rightarrow B$ \leftarrow دى معناماً بـ B تحدى وجوهها على A
 => $A \rightarrow B, C, D$ \leftarrow اد ايميلو دى علاقة بين A و B, C, D \leftarrow Functional dep
 => اد table يبلوون سالم لو كان يتحقق العلاقة دى
 * يعني اد A هو اد PK وار PK \rightarrow B, C, D باق الدالس \leftarrow لو عفنا به طرق اد
 * اخبي باق الدالس كدا العلاقة سالم

→ keys and dependencies:

$\rightarrow Emp(Emp_ID, name, age, Salary)$

A diagram illustrating a functional dependency. A box labeled "determinant" contains the attribute "Emp_ID". Three arrows point from "Emp_ID" to three attributes: "name", "age", and "Salary", which are grouped together in a box labeled "functional depen".

* ملائیں تکنالوژی میں لازم کل اور Columns تکنالوژی اور Pk بس

\Rightarrow Types of functional dependency:

① full functional: راہے columns مکمل اور composite pk: a^ن table کے
 {SSN, Pnum} → hours full func العلاقہ values جوکی composite pk اور

② partial func: لوف حزء بعثته يعنى جزء من ال PK هى كل ما يقتضى
الحالقة partial يعنى لارم فى الاتهادى يكون ال PK
كلى فرار full عددى يكون simple

③ Transitive func: المُنْتَهِيُّ بِالْمُنْتَهِيِّ، keys و columns بِوَفِيَّة ←

\Rightarrow steps in normalization:

∴ Zero normal form \Rightarrow ↵

also modifications) gives rise zero normal if it is also table if it is not

- ونحوه هو نوع من أنواع الـ normalization أو الـ normal form (الـ NF) -

بِرْ تَوْلِي table جا

: first normal form ←

1st normal form (1NF) ↳ repeating groups ||, multivalued Attr || similar to table || -

- وتحل دایا ناخد ار columns as مجموعه repeating groups للتبی انداز تکرار یافته

Composite key \Rightarrow key of table

: Second normal form ←

- هو دليل partial dependency لـ second normal form -

جزء من table يتم تعيينه مع اطاحة الى يحتوي على primary key و foreign key

: Third normal form ←

- هنوز مدخل لل third نسبية الابتدئي transitive dependency الابتدئي third table ←
لوحة table الابتدئي third keys، three columns

Ex:

(Eid, Ename, Eaddr) | (Pro_id, Pro_name, Pro_loc, Pro_city, hours, Bouns)
repeated cols ← repeating groups ←

: 1st N علشان نوصل لل ←

Project الابتدئي 2 tables ← نعمل الجدول الجدول ←

→ Employee (Eid, Ename, Eaddr) composite key ←

→ Emp_Proj (Eid, proj_id, Pro_loc, Pro_city, hours, Bouns)

: 2nd N علشان نوصل لل ←

هستوف لواز 1st نفع 2 tables ←

composite key ← هستوف لواز partial PK فيه single PK ←

هيلون في المديروال city بقى project بعنوان Proj_id

→ Employee (Eid, Ename, Eaddr)

→ Project (Proj_id, Pro_name, Pro_loc, Pro_city)

→ Emp_Proj (Eid, Proj_id, hours, bouns)

: 3rd N علشان نوصل لل ←

transitive لوقت ←

أول جدول مفيس وجدول الـ Emp_Proj

الـ 2nd جدول الـ project فيه الـ project city وال loc

→ Employee (Eid, Ename, Eaddr)

→ Emp_Proj (Eid, Proj_id, hours, bouns)

→ Project (Proj_id, Pro_name, Pro_loc)

→ locations (Pro_loc, pro-city)

⇒ 4-SQL Aggregate func, Grouping, Union, subqueries, EERD

① ⇒ Aggregate Function:

built in functions

Count, Max, Min, Avg, Sum رأى *

أى يطلع منها Aggr func اى Value موجودة في table Query

↳ Select Sum(salary) from employee table قيمة مجموع ال

query يطلع استخدم أكبر من func المطلوب

↳ Select Min(salary), Max(salary) from employee

: table عدد rows العاشر

↳ Select Count(Eid) from employee count(*) *

Null rows المفقودة Count المعمول PK column اي Count

↳ Select Count(ename) from employee

برضوا 15 rows لوحدها NULL

كما مجموع ال 15 rows

↳ Select Avg(salary) from employee

لكلها هنقسم على الأذن اعتدن NULL=0

② ⇒ Grouping: groupby query

↳ Select Min(salary) did كذا اللي هيجله هيسنو ال did وكل مجموعة لهم نفس ال did بتعتبر group ويبيهم

group by did

ال results min(salary) وبستوف لو فتح شوية تاني لهم نفس ال did

جبلهم ال did min تابعهم يعني الناتج هو تكون حسب id did

فتشي 5 مرات و did=20 تكرر 3 مرات يعني did كل group

لهم min يعنى هيدلهم 3 نوافع وهيلتون يعني جدول فيه 2 columns

فيه قيم ال min دوامه لل did و مكتوبين صواب ال rows متوجه الناتج

↳ Select Count(Eid), Address Cairo & All Address كل ال addresses بحسب address

6	Cairo
5	Alex
4	Mansoura

صلباً بعدهم

↳ Select Count(Eid), Address

: where مع group by

from employee

الاوليند ال 3 rows هو هؤلا

where did in(10,30)

mansoura, Alex, Cairo

group by Address

بس ال هنقدر عمال value بتاعت ناتج

عن هنقدر بس الناس اللي لها did 10 او 30 لكن لو

ليه did مفهمل موجود في الجدول

↳ select Min(salary), did
 from employee
 where address like '%
 group by did

* بفرض أنك تطلب مني إيجاد الأدنى للأجور
 * بفرض كم did هي المجموعات التي تم
 تجزئتها في المكان حيث
 تجزئ المديرين في القاهرة
 المديرين في المنيا
 المديرين في الإسكندرية

→ Having:

↳ select sum(salary), did
 from employee
 group by did
having sum(salary) >= 2200

* ما ينفعنا نستخدم where مع Aggregate function
 * وما ينفعنا أن where تتحقق بعد group by
 * علىikan كذا استخدمنا بالها having
 يعني كذا معناه يفلتر الأقسام حسب did تأثيرها
 وعند كذا كل group ليه نفس ال did فحسب الـ sum بتاع الـ salary ، يعني
 اللي الـ sum بتاعهم أكبر من 2200

↳ select Count(Eid), Address
 from employee
 group by Address
having Count(Eid) >= 5

* هيفقسم الـ table لـ groups حسب الـ Address
 * بعد كذا كل id فيه دايم Eid مثلاً Cairo
 فيها 5 id لها فيها 5
 فيها خداو 5 id لها فيها 5
 هيا خداو 5 id لها فيها 5

5	Cairo
5	Alex

ow alex , cairo , ca

↳ select sum(salary), did
 from employee
 group by did
having Count(Eid) > 5

* من لازم الـ select ليكون هرمه اللي
 having in
 * يعني هنا هيفقسم الـ table حسب الـ did
 group all salary .
 بعد كذا هيجيب .
 يعني having بس الـ group اللي فيه عدد موظفين أكبر من 5

↳ select Sum(Salary) , did
 from employee
 where address like '%
 group by did
having sum(salary) > 12000

* هيفستوف الـ address ، اللي فيه تأثير حرف
 alex بس الـ did Mansoura و Cairo
 * هيكل groups حسب الـ did 30,20,10
 group all salary .
 كل group هيكل group .
 بس Mansoura و Cairo

12000 يبرهننا ان الـ sum الـ groups

↳ select Max(salary), address
 from employee
 where did in (10, 30)
 group by address
 having Count(Eid) > 3

↳ يبحث عن الأقسام التي لها 3 أو أكثر من
 موظفين أو أكثر من 3 موظفين
 في نفس العنوان

① ↳ EX:
 ↳ select count(*), count(st_id), count(st_iname), count(st_age)
 from student

↳ ي RETURN 4 columns

↳ selects rows if the value of st_id is PK * or the count of rows
 null values for st_age, st_iname etc

② ↳ select avg(st_age) from student

↳ calculate the average age

③ ↳ select avg(isnull(st_age, 0))

↳ ليعاير ال rows كلها

↳ ويعتبر null values as 0

④ ↳ select sum(salary), d.dept_id, dept_name
 from Instructor inner join Department d

↳ عندي 2 tables

↳ group by d.dept_id, dept_name

↳ calculate the average salary for each department

⑤ ↳ select avg(st_age), st_address, dept_id
 from student

↳ group by st_address, dept_id

↳ calculate the average age for each address

⑥ ↳ select avg(st_age), st_id
 from student

↳ group by st_id

③ ↳ **Subqueries:** query input query output query

↳ select * from student

↳ select * from student

↳ where st_age < (select avg(st_age) from student)

- m → select *, (select count(st_id) from student) from student
- ← كل الأسطر في student مع كل
 ال rows في subquery
- m → choose table name instead of inner query or outer query
- ← لازم الـ aggregate function مع
 لازم الـ subqueries
- m → select dept_name from Department
- ← اد سطيب او قام الاقسام
- where dept_id in (select distinct(dept_id) from student where dept_id is not null)
- ← المقصود بالـ null في الـ predicate
 علشان كل الأسماء
- in = لأن الـ null يرجع كـ value من نوع واحد وله قيمة
 وبعد كـ اهتم من الأقسام الـ null رجع من الـ inner query
- Department is outer table student is inner table
- ← وهذا الـ result يتسلق على الـ inner query
- m → select distinct dept_name from Students inner join Department d
- on d.Dept_ID = s.Dept_ID
- ← الـ join أسهل وأحسن صيغة
 join ↓ subquery engine

⇒ Subquery + DML :

m → delete from Stud_Course
where st_id = 1

m → delete from Stud_Course
where st_id in (select st_id from Student where st_Address = 'Cairo')

← هنا هيبي ألقا الطيبة الـ 5 في الـ Cairo

وتحتاج مسح جدول تابع

④ ⇒ Union :

m → select st_fname
from student
union all

select ins_name
from Instructor

distinct

union

select st_fname

from student

union

select inst_name
from Instructor

join table 2 على نفس بعد

* لوماحظنا union all في النص حيث كل row لها

وناتجها يبقى خل الـ sheet لكن كلهم يستطيعون تعيين نفس

اسم الـ column هو تم اول حاجة الـ user

أغيره عادي

* لازم بعد الـ column فـ order select يعني الـ 1 الثانية

union all معه union ←

union all معه union +

* هنا هيبي الـ unique names والـ address

* وفرضوا لازم كل cols يكون زي بعض وتابع

ارـ data UNI data فـ انظر

مع بعض

↳ **Intersect:** distinct كـا هـ يـ بـ اسـمـاـنـ الطـلـبـةـ الـمـشـبـبـ بـ اسـمـاـنـ الطـلـبـةـ الـمـشـبـبـ

 ↳ **select st_fname from student Intersect** بـ رـضـوـ بـرـتـيـبـ وـتـسـيلـ الـكـلـيـنـ بـعـدـ

select ins_name from Instructor بـ رـضـوـ بـرـتـيـبـ وـتـسـيلـ الـكـلـيـنـ بـعـدـ

 ↳ **Except:** distinct كـا هـ يـ بـ اسـمـاـنـ الطـلـبـةـ الـمـشـبـبـ بـ اسـمـاـنـ الطـلـبـةـ الـمـشـبـبـ

 ↳ **select st_fname from student Except** بـ مـحـاـهـاتـ الـرـقـمـ الـأـذـوـيـ وـمـسـخـ الـتـابـيـةـ

select ins_name from Instructor بـ عـلـىـ بـطـرـجـ الـكـلـيـنـ بـعـدـ

distinct بـعـدـ **Except**, **Intersect**, **union** كـا

EERD (Enhanced ERD) → (ERD + inheritance)

post_stud, under_stud صـنـفـنـ النـوعـ نـارـ 2~ Entities

 بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ مـلـكـ مـصـرـكـ الـنـوـعـ

 بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ مـلـكـ مـصـرـكـ الـنـوـعـ

 Name وـ ID

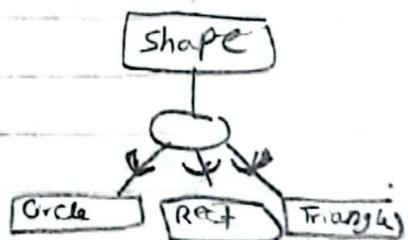
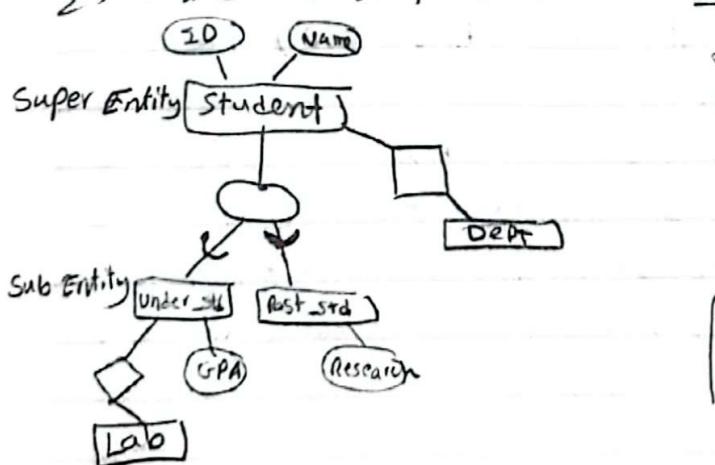
مـلـكـ مـصـرـكـ الـنـوـعـ مـلـكـ مـصـرـكـ الـنـوـعـ

 student لـئـلـ super Entity

Sub Entity بـعـدـ post_stud وـ under_stud

inheritance بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

business, post_stud is a student, under_stud is a student



Sub Entity بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 generic بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 super general بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 specialization بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 Super فـارـ

generic بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 super entity بـعـدـ مـلـكـ مـصـرـكـ الـنـوـعـ

 لوـحـدـهـ كـانـ

+ لما نعمل على relations لون حايدة ليها علاقة بكل الـ subEntities هنربطها
هرة واصحة بـ superEntity بـ س

⇒ Constraints in supertype:

① Completeness Constraints

- Total specialization Rule : Yes (double line)
 - partial specialization Rule : No (single line)

② Disjointness Constraints:

- Total Disjoint (d)
 - overlap Rule (o)

لوار ناچرل سپر دبل ٹائی دی جن اسے \leftarrow اکی تھیں جو دوں بس مخفیں غیر رسمی

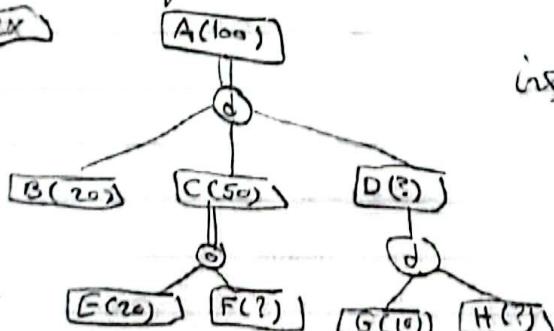
بتویه generic as sub Ent و من هنکه مع از sub کل partial و single like است

→ Discriminator (disjoint rule) see video 2:18; 2:25
(overlap rule) 2:25; 2:29

→ Transforming EER Diagrams into Relations: (Mapping)

وكلمة Super Ul Table ، super Ul table \leftarrow
Super Pk \leftarrow Primary Key \leftarrow PK \leftarrow Primary Key \leftarrow super Ul

ex



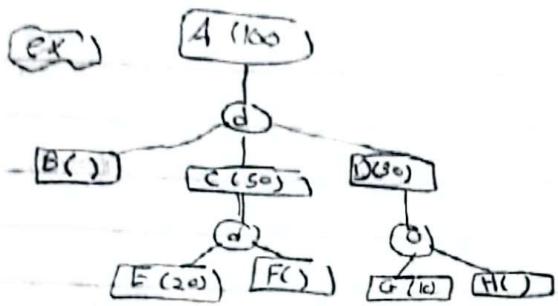
$$A = B + C + D \rightarrow D = 30$$

Partial تجزیه داری

$$\therefore \alpha \leq H \leq (\Delta - G) \Rightarrow \alpha \leq H \leq 2\alpha$$

overlap & partial ω & π

$$30 = \angle F \Leftrightarrow 50$$



$$\begin{aligned} & \therefore 0 \leq B \leq 20 \\ & F = 30 \\ & 0 \leq H \leq 30 \end{aligned}$$

→ Select st_fname, st_age, dept_id
from student
order by 1
→ Select st_fname, st_age, dept_id
from student
order by dept_id, st_age
asc desc
→ delete from Department where dept_id=20
→ update Department set dept_id=3000 where dept_id=20
→ built in functions:

- * Agg funcs: getdate(), isnull, coalesce, concat, convert
- Select year(getdate()) → Select month(getdate())
- Select substring(st_fname, 1, 3)
from student
- Select db_name()
- Select suser_name()

→ view → template explorer

* فيه كم حاجة فتحتها ، إزاي تفتح

⇒ 5: SQL, DB Engine, SQL Services, Ranking Func, Transact SQL

→ Overview of Relational DB: MySQL, Oracle, PostgreSQL, SQL Server
fully relational DB engine

M-M, 1-M, 1-1 (all cardinality is allowed) DB
lets tables have 1-M yes (into RDBMS or Physical) or int *
child, parent as ميزة

Microsoft SQL Server edition, SQL Server Version
الفرق بين Microsoft SQL Server tools and version *
file features من edition *
لما ننزل SQL Server

SQL Server Management Studio, App, Service, Instance → Setup File *
SQL Server Management Studio, App, DB Engine (SQL Server (MSSQLSERVER)) Service *

→ Security:

* authorization (permissions).

* authentication (UserName & pass)

authentication (نوعين ones SQL server) ←

SQL Server authentication windows authent

→ top:

→ Select top(3) * from student

كل 3 rows هم اول 3 rows

→ Select top(3) st-name from Student

هم اول 3 rows هم اول 3 rows

→ Select top(3) * st-name

alex is 3 rows هم اول 3 rows

from student where st-address = 'alex'

→ Select top(3) salary from instructor

3 rows اول 3 rows اكبر

→ Select max(salary) from Instructor

salary هي اكبر

→ Select top(2) salary

اللذين زادوا اعلى قيمة لهم

from Instructor order by salary desc top(2) ← Order by

→ top with ties:

→ Select top(3) with ties * ← هم اول 3 rows هم اول 3 rows

from student

لدين تالت row الاول st-name هم اول 3 rows

order by st-age

تالي بغضون هم اول 3 rows هم اول 3 rows

`newid()` (global universal ID) GUID یا مخفی func is #

Server will generate unique & random (جديد) ID every run لذا كل مرة

→ select top(3) * from student and الستاد لـ 3 اولى في ترتيب النهاية

• from student order by newid() top(3) 1st, newid() last, randomize rows

* دادا مقتدی ف حالتہ ما اکون عایز لخت دیم ٹھوائیں

→ execution order:

→from →join →on

→ where

→ group → having [Aggreg]

→ select [distinct + Aggreg]

→ order by

→ top

→ Select st_fname + ' ' + st_lname as fullname
from student
order by fullname

↳ Select st_fname + ' ' + st_lname as fullname join @ two rows will
from student join two rows will
where fullname = 'ahmed ali' select all

→ select * : Subqueries für Union

From (Select st_fname + ' ' + st_lname
 Label student subquery) WS *

الدول ملحوظات كـ (الإجابة) (as fullname from Student) as newtable

where fullname = 'ahmed ali' fullname is also in where it

DB objects: [table , view , function , SP(store procedure) , Rule]

: and set your default path and also give links DB 11 & object (S) * a

→ [serverName].[DBName].[schemaName].[objectName]

* الـ object خارج الـ VB يسمى بـ (new) لـ create object

objectName is student, schemaName = dbo

کل داس لازم تر مدار object ہے اور table کی وجہ پر دوں راہی خیال کریں۔

ـ كابيغيل union لاذقسام الـ
ـ موجودة في الـ companyso cust DB
ـ الاقسام الموجودة في iti

```
→ Select dname
  From CompanySD.dbo.Departments
  Union all
  Select dept_name
  From Department
```

→ Select into (DDL) :

→ `Select * into table2 table1 Create جملہ کا یہی DDL نہیں`
from student `سے اسی جملہ کا یہی table2 کا جو جدول ہے اسی table`
`table Student نہیں بلکہ table2`

→ Select * into CompanySD.dbo.student ف نے table I create JE's
From student ↪ student اکی کا نئی جگہ LMS تک DB

→ Insert based in select;

→ insert into tab3
values (66, 'ali')

insert into tabs طلب Student موجود بالموبايل table no data available
 select st-id , st-fname from student tab3 table فاني يرضي موجود بالموبايل
 st-id || student lines will be inserted into tables of structures
 tab3 table فارجع st-fname

→ having without group by:

→ عادی ترجیح لوردها بس لاتم از having sum(Salary) from Instructor select
→ يکوں فنیها Aggregates Cols جیسا Count(ins_id) < 100
→ یعنی دا هیطلع جمیع اړلزبات سشرط نهدا المونټین یکون اقل من 100 راتون هستونو فی الدول
→ نهدا المونټین قبل ماتدل sum

Ranking Functions: alias for special built-in functions *
Business logic

- Row_Number()
- Dense_Rank()
- NTiles(Groups)
- Rank()

* يعطي ترتيب الارقام حسب الامر المخزن
الا، قام بـ تنحیط ارزی، وشوندگی الارقام ارزی
في الاتجاهية

→ Select *, Row_Number() over (order by esal desc) as RN,

Dense_Rank() over (order by esal desc) as DR,

From Employee

col RN over جملة over sum of all rows

eid	ename	esal	did	RN	DR	G
10	ahmed	10000	10	1	1	1
9	ali	10000	10	2	1	1
7	eman	9000	10	3	2	1
1	nada	9000	10	4	2	1
2	reem	8000	20	5	3	2
3	khaliq	7000	20	6	4	2
6	moham	7000	20	7	4	2
5	suley	6000	30	8	5	3
4	hasan	6000	30	9	5	3
8	omar	5000	30	10	5	3

Rank function over order by esal desc

* الـ RN تقبل الـ RN كل الدول كل العمال

ومرتبهم تنازلي حسب الـ RN كل العمال

او eid ظاهر من الترتيب

بعد كل col comes RN وناتي فيه ا، قام

او rows بالترتيب

* او (RN) over order by esal desc Dense_Rank()

بعد كل col comes DN والـ RN فيه ترتيب فيه *

ا، قام او rank وكل مجموعة متساوية في او لـ RN يعني الـ RN

10000 لها صدو رقم 1 وكم عدد زملائهم وكم عدد زملائهم باضو رقم 2

RN can error from where RN=1 لـ RN error

Subquery لـ RN like where RN=1 شرطه بعد الاو where select alias 1 =

→ Select *

From (select *, Row_Number() over (order by esal desc) as RN,

Dense_Rank() over (order by esal desc) as DR,

NTiles(3) over (order by esal desc) as G,

From Employee) as Newtable

where RN=1

top لـ RN like where RN=1 top لـ RN like where RN=1 subquery like max(RN).

sal مسنه يتحقق غير الطريقة دي لو نايزين RN=3 مثل دي كذا ياعيز قالتن اعلى

لو قال where DR=1 كذا يجيب كل او rows الى ليها اعلى مرتبة مثل هنرجع

row واحد هنرجع لهم الـ RN

\Rightarrow partition by :

→ insert into ...

select *

from (Select*, Row_Number() over (partition by did
order by real desc) as RN,

DENSE_RANK() OVER(PARTITION BY did ORDER BY esal
DESC) AS DR

from employee) as new table

fid	Chname	esal	did	RN	DR
	ahmed	10000	10	1	1
	ali	6000	10	2	1
	eman	5000	10	3	2
	hada	8000	10	4	3
	Reem	17000	20	1	1
	Khaled	6000	20	2	2
	makame	6000	20	3	2
	Sayed	15000	30	1	1
	hussen	4000	30	2	2
	omar	3000	30	3	3

لما ندخل group قبل ار order دى ناري ار partition by بس الفرق بين group و partition ان ار group يتحقق اى بعدي ار partition و same partition اى بعديها المترتب على الـ all بس

did the groups include order of partition II +
order 3a, order 2a, group 1 did not yet come

لہ کا دل partition order میں ہر سب مرتبات +
القسم 10 وچہرے بین 11، 12، 13 و 30

* اگر $RN = 1$ like where $RN = 1$ like $\frac{1}{2}$ کا میانہ سینا کے ایک مرتبہ خیل کل قسم و کانوں کا partition برضو هستئے علی کل (لوگوں) Dense All Row-Mumber

روزی بس group by مع max لایه هایی باقی میگیرد

* DR = 1 درجة عابر اي صرف في كل قسم ٥٠ الف لتر ار

کل لو ۸۰٪ ریس ایج صربیل می می خدمت مع اسکر

⇒ Data types

→ Numeric DT:

→ bit → bool 0:1 True; false

→ tinyint → 1Byte -128:127 unsigned 0x255

→ smallint → 2B -32768: 32767 unsigned 0:65535

→ int → 4B

→ bigint → 8B

→ Decimal DT:

→ smallmoney 4B 0.0000 4B يأخذ 14 رقم العادمة العشرية وال الصحيح

→ money 8B .0000 يرضو 4 بعده العادمة

→ real .0000000 العادمة 7.

→ float .000 — 0 يأخذ range كبير

→ dec decimal dec(5,2) ال dec أكثر واهر خال استخدموه

منهاه الرقم متكون من 5 digit منهم 2 في العادمة يعني منك 123.45 أو

12.3 عادي يرضو لك 12.345 ما يقتصر مع انه 5 بعده

العادمة ما يقتصر يكون أكثر منه 2 digit

→ Char DT:

* دا حات لو كيست كلة 3 حروف هاي حجز رضوا ال 10B ومن سيف غير انجلزي

→ varchar(b): variable length هيجبر على حسب المطالعات اكتب ويرضوا من هنكله غير حروف انجلش لو

حروف عربي هيظهر ???

→ nchar(10) : unicode دى هي fixed برضوا بس تعلم الخطة وتحقيق عربي

→ nvarchar(10) : برضوا هيفهم عربي بسلسلة variable length

→ nvarchar(max) : up to 2GB ال الوصل عارف فال String قيمة كام هتفعل تعلم 2GB

→ DateTime:

→ Date MM/DD/YYYY

→ Time hh/mm/12.765 بعذر الثاني بالثانويون هنا دا ال برج 13 من الثانية

→ Time(7) hh/mm/12.1234567 دى بدل اعماق اعماق يعني كم ال جزء من الثانية

وذلك تعلم اعماق عماق time(7)

→ Smalldatetime MM/DD/YYYY hh:mm:00 دا صحن بيعرف الثاني بخلينا صفر ويرضوا التواريخ القديمة صحن بيعرف بيتعامل معاها

→ datetime MM/DD/YYYY hh:mm:ss.123 دا صحن بيتعامل تابع الثاني والتاريخ الفريدة

→ datetime2(7) MM/DD/YYYY hh:mm:ss.1234567 دا طبعوا لا يلي accuracy

→ datetimeoffset : 11/24/2020 10:30 +2:00
محلية جداً التاريخ والوقت
و ساعتها سوقية القاهرة

→ binary DT:

→ binary 011101

* يخزن 1,0
* يخزن الصورة في DB وهذا حسن سأنا نخزن
ال path بناء الصورة في حالة أي يطلب backup لا DB كيتر بس لما بتغير
هذا يخزن كل سار DB بس هنكون معاً طول

→ others

→ XML

→ unique_identifier

→ sql_variant

⇒ Case when end

update, select يستخدم مع *

→ select ins_name, salary,

case

when salary >= 3000 then 'high sal'

when salary < 3000 then 'low'

else 'no value'

end as Newsal

from Instructor

→ update Instructor set salary =

case

when salary >= 3000 then salary * 1.10

else salary * 1.20

end

⇒ iif

false الأول، true الثاني و false كيتر 2 condition لـ learning *

→ select inst_name, iif(salary >= 3000, 'high', 'low') from Instructor

⇒ Convert, Cast, format

→ select cast(getdate() as Varchar(20))

string لـ date

→ select convert(varchar, getdate(), 102) مع نوع

or date يوضع مثل الـ date أو dateformat كـ parameter

→ format (getdate(), 'dd-MM-yyyy') الـ date > الـ date في format *

format (getdate(), 'hh:mm:ss')

- select format(getdate(), 'dd') دى هترجع اليوم على اكسل string out
- select day(getdate()) دى هترجع على اكسل int out
- select eomonth(getdate(1)) برجع آخر يوم في الشهر end of month
- select format(eomonth(getdate()), 'dd') علامة format و علامة
- select format(eomonth(getdate()), 'dd') علامة ذكر آخر الشهر 31, وهو
- select eomonth('1/1/2000') هرجع 31 الى صورة آخر يوم في شهر 1

Day 6

\Rightarrow 6- SQL Database Constraints, Rules - create OB \Leftarrow

creation of DB will be done by DB engine & creation of table by microsoft sql DB & physical representation (1) SQL Server's own data levels : logical Represent (2) meta data file : physical files & (1)

relations لی tables لی وہ فہرست meta data file Mdf * transactions لی وہ Ldf * SQL کے create as File Group ②

Mdf pointer to its primary filegroup *
 Mdf Leaf secondary files 1.2
 عاليات مفهومها files ستاوريها ضد F92 و F93 2.2
 يشار على الملفات physical files أو أكثر

كل file .Ndf يملئن بـ space مفتوحة على الhard disk ←
 # ل الواحدة Data pages كل 32kb هي تخزن على صفحات
 PK و تقويم صرامة اى Page = 8kb

→ اد derive Attr المفروض معن بخطه او table من خطه في الـ table لو
الـ equation المنسوبة لها كانت طيبة ومتاخر وقتاً ، أول لـ Attr دى متحصل
في حساب حاجة تانية وكل دا صيارة لـ ~~Attr~~ Performance

table یا derived Attr یا طبقه ایکیا ایکیا Microsoft فر
computed column یا formula یا Attr یا کسی ایکیا table کی properties یا وہ
Attr یا کسی yes لئے is persisted یا ایکیا ایکیا formula یا

* دی لوف حاجہ NULL میں چکبر (Null) میں
Yes lastie IS sparse (sparse property) \leftarrow

: Diagram خال relations بحسب properties ←
 طاينجی هنسج child و parent میں کافی سو بینج ←
 * دلایت ن فرugal properties ←
 update Rule , Delete Rule ←
 * دلایت ن فرugal properties ←
 No action
 Delete کا اعلان No action ← set NULL like Delete Rule ←
 * لوگونا ال NULL بے FK ← خال ال
 * و تھو ار Parent child میں cascade لے لیں update Rule ←
 لے لیں cascade child و امیر و میں delete Rule ←
 کافی ان طبقہ child parent را فر جاتے ال
 dependent child child parent ←
 dependent child child parent ←
 * لوگونا ال weak entity Employee ←
 * لے لیں ار default value ← set default value ← Delete Rule ←
 لزム اکون یافت ار property ←
 و هنچاها نہ ار default value ←

⇒ schema Name

dbo.Student ← schemaName . objectName میں object ←
 اسی ار اخترعت ار میں تھل 3 میں اکل :
 2object ←
 * اول میں table ایسا میں تھل اعل 2tables بینس الاسم فر ار DB ار
 * میں تھل table لے لیں tables کیترف ار DB هر وکیس والاحسن کنت اعل جروب
 * کل ال logical grouping ایسا جو ایسا جو ایسا جو ایسا جو ایسا جو ایسا جو
 * لوحہ عایز اعل table permission اکل table ار user ار pass
 * اکلول الی علیہ ار schema ار
 * هقر ر اعل 2tables میں 2tables ←
 * tables ال logical grouping ار schema ←
 * table اکل table بے user ار schema ←

→ Create schema HR DB موجودہ ار HR Ifaw schemas ←

فی قائل ار جو اہ قائل HR schemas ایخط و security ←

→ alter schema HR transfer Student ← کافی ار ایسا جو اہ قائل ←

بیت اسیم HR . Student ←

→ alter schema HR transfer Instructor schema ← کافی ار ایسا جو اہ قائل ←

→ Create table student ← کافی ار ایسا جو اہ قائل ←

(id int , ← و پیخت ن ار default ایسا جو اہ قائل ←
 name varchar (20) ←

)

→ create table sales.student (→ schema) create table student
الى يائزه يتحط لـ login طول

→ لازم كـ اخطـم الـschema الـHR.Instructor default لـ HR

→ هيـعاملـع الـيـ موجود فـار HR قـدـمـهـ

→ خطـوات الـ HR عـلـشـان نـفـل pass , username) create security

update delete insert,select,logon : (نـفـل user server login

→ اـولـ حـاجـةـ properties server rightclick وـنـفـلـ الاـخـتـارـ ①

SQL and windows authentication هوـ المـصـوـرـ mixedmode

restart, وـنـفـلـ rightclick rightclick server. I restart ②

→ جـواـفـاـيلـ اوـ security فيـلـ اـسـافـلـ وـنـفـلـ login ③

وـهـنـلـكـ Enforce pass policy وـنـفـلـ password وـنـفـلـ Login Name فيـلـ user

→ جـواـفـاـيلـ DB الىـ يـائـزـ الـuser فيـلـ login هـنـجـيـسـ فيـلـ security ④

وـبـوـاهـ فـاـيلـ new user وـنـفـلـ rightclick وـنـفـلـ login وـنـفـلـ new user

مرـتـنـ فيـلـ SQL Server كـ ماـنـقـحـ الـuser login name وـنـفـلـ user

وارـ pass دـولـسـ هـنـجـيـسـ DB بـسـ الـيـ عـلـيـهـ new user

اوـ قـيـمـهاـ

create schema ⑤

→ خطـ مـعـوـدـهـ الـalter schema HR transferstug schema هوـ الـ objects

schemas الـ DB security منـ فـاـيلـ user وـنـفـلـ schema

ـنـفـلـ الـ doubleclick وـنـفـلـ search وـنـفـلـ permissions

اختـارـ Browse وـنـفـلـ user

→ insert & select منـ خـتـمـهـ Grant وـنـفـلـ permissions ⑥

update وـnـfـlـ delete وـnـfـlـ select وـnـfـlـ Deny

ـنـفـلـ disconnect وـnـfـlـ connect وـnـfـlـ user (حقوقـ user فيـلـ server)

ـنـفـلـ) وـنـفـلـ Schema tables يـائـزـ الـ connect ,

ـنـفـلـ DB الىـ يـائـزـ new query وـnـfـlـ permission run

ـnـfـlـ insert into HR.Stud (st_id,st_name)values(--) ;

ـنـفـلـ هـنـجـيـسـ

ـnـfـlـ delete permission error (update HR.Stud set --)

Synonym

→ select * from HumanResource.EmployeeDepartment schema HR+ لـ
 راج table كبير ولو استخدمناه في funcs ، حاجات كثيرة وحيث أن اسمها يأخذ
 كما هي كما في غيره فكل حاجة استخدمناه فيها

→ Create synonym HE
 for HumanResources.EmployeeDepart shortcut و لو جزء
 change name shortcut to alter name shortcut كل حاجة استخدمناه فيها ، وهنالك
 change name shortcut to alter name shortcut هو فايل ال Synonyms

drop , delete , truncate

→ drop table Course no metadata وال data وال delete
 ال table فيه ال DB

→ delete from Course

→ truncate table Course

← العرق بين delete و

* لومايز (لغز جزء من ال rows هستمز where من يتغير معها

* ال delete أبطأ منه truncate حوا ال loc file

ال truncate من دارما تكتب حوا ال loc file

* حذف كل row back ورجح ال rows ال المتبقية

ال truncate ما يغير ارجع في تابع

* identity reset لـ delete table identity يعني لو كان table فيه identity من 1 to 100

وكلما delete r insert r على id تكون ال id يـ 1001 من حيث انهم

1 reset يـ truncate all كـ 1

* لـ delete row فيه وـ ad PK table parent لـ 1

truncate

table لـ rows structure column مـ note

NULL to column U set table update

DB Integrity

Constraints

أو rules constraints between tables بينها tables مفهوم DB integrity *
باحتام الاراده الالى فيها

* DB integrity يعني ان DB لا يمكى حفظه ومتقابلة rules وارules تحقق السينس

- Domain Integrity

و Davies طريقة اى تتحقق 3 حاجات هم:

- Entity Integrity

تحقق ال 3 حاجات دون عن طريق:

- Referential Integrity

DB objects - و DB constraints -

: DB Constraint ←

: Domain Integrity ①

columns لـ range of values بين Domain *

* الى يتحقق ال DB ملحوظات اهمية دا : Da range of values

* او Data type : لا يمكى بضمون نوع البيانات او زرائى int لـ range

DB يتحقق بمحاجة معنوياتي واصحة من القيم الممكنة تكون في او لـ Default value

DB يتحقق من القيم التي يقع تحظرها في او لـ Allow NULL , Not NULL

DB يتحقق من خواص الـ table creation where Check constraints

أحد او age مثلاً من 20 لـ 50

: Entity Integrity ②

* يتحقق ال rows لـ uniqueness

و دا يتحقق NOT NULL و دا PK constraint

Allow NULL يتحقق اى ادخل unique key : unique constraint

اكيد NULL و اى اى و اقدر ادخل أكثر من اى ي تكون unique

: Referential Integrity ③

* يتحقق ال relationships بين او tables تكون مطبوبة

: Foreign key constraint

: DB objects ←

* يتحقق ال DB integrity على مستوى او table ك الخاص بالtable

table Fk , unique , pk , check و دا يتحقق على مستوى او table

Triggers او Rules

: Entity Integrity ②

Triggers او Index

: Referential Integrity ③

Triggers

* كما انواع الـ triggers على DB constraints

labeled script (ويكتب كود او Custom constraint) unique constr , Fk constr

عشر وظيفة

Constraint C7 check overtime between 100 and 500,
 مدار الوظيفة C7
 Constraint C8 foreign key (did) references dept(dept_id)
 on delete set NULL سطواري از اين الک یعنی dept_id از table dept
 on update cascade دفعه متوجه حاله از update اينها والي يحصل عليه
)

→ alter table emp add constraint C100 check (hour_rate > 100)
 بعادي اعل از table مع constraint مخصوص اينها بيس لازم الشرط يكون يعني
 عالي البيانات القديمة والجديدة يعني هنا كامل الا لإدخال hours_rate<100 وحاله الاتصال قيمة في
 كل شرط كاملاً صيرضاً ي العمل ال constraint

→ alter table emp drop constraint C3 بعادي اعل از table مع constraint
 مساله ال

→ Rule
 ↳ لوكيز اعل constraint بطبقه على البيانات الجديدة بيس ② وعيزيز اعل
 datatyp e ③ بقوت بين کما shared table بقوت constraint
 و Constrains default و ملحد ياخده يلوون عليه ال automatic default

اد Rule بيعتبر مستوى ال DB ④ او schema ⑤ بعدي
 ال variable بعدي ⑥ : ⑦ ال

→ create rule r1 as @X>1000 rules اد rule کا الکها create rule
 جوا جدول از programmability

→ sp_bindrule r1 'instructor.salary' ⑧ stored procedures SP
 اد rule کا الکها sp_bindrule از runtime و کتب کتابتها از
 Instruct. Salary

→ و کذا ال rule هستيماً تطبق على القيم الجديدة الکها للن قيم زرا فما هو
 بيس لو ندلات على القيم برمجنو تطبق عليه ال rule و کذا حفظت رقم ①

→ sp_bindrule r1, 'emp.overtime' نفس ال rule طبقناها table کا و کذا حفظت رقم ②

→ sp_unbindrule 'Instructor.salary' اقبل ما اتعل rule اذ م
sp_unbindrule 'emp.overtime' bind rule کا طاجة
drop rule r1 اتعل rule

constraint	rule
is مستوى او table	کذا الفرقین
specific	DB او rule بمستوى ال DB
الاتا القديمة والجديدة	shared
Constraint مع آخر على	تتحقق على الاتا الجديدة بيس
آخر	1 rule کا عمل عليه

→ create default def1 as 5000

SP_binddefault def1, 'instructor.salary'

SP_unbinddefault def1 'instructor.salary'

programmability → Defaults اور file فی create لاما

drop default def1

عادی def1

bind rule و نسل

unbind

و کما آنکہ

→ Create Datatype

جیسے datatype جیسے

ComplexDT (int > 1000 default 5000) is

→ create rule r1 as @X > 1000

default و rule

create default def1 as 5000

SP_datatype ComplexDT, 'int' after int و now ComplexDT

programmability فیا Type فی ال User-Defined Datatype من Create

بس کہ اور rules و defaults میں دو بیوپین یعنی Datatype اور

→ SP_bindrule r1, ComplexDT rule نے bind کیا

SP_bind default def1, ComplexDT ComplexDT اور ایک datatype کے default

لئے 5000 as لے default 1000 وار int کی قیمت تکریبیں اور

→ create table test

کہ اور salary اور rule کے سطح پر ایک constraint

(id int,

default=5000 و یا خاص loca int

name varchar(20),

salary ComplexDT

) rule کا ایک constraint کے طبقہ کیا جائے اور rule کے طبقہ کیا جائے

هر فہرست میں rule datatype اور rule کو نہ کہا جائے

datatype میں ایک constraint میں rule کے طبقہ کیا جائے

constraint میں rule کے طبقہ کیا جائے

check constraint میں rule کے طبقہ کیا جائے

7- SQL, variables, if, while, Functions

1 Variable

: local var ①

* يمكن ايجاد local var في مستوي batch او stored procedure او function

→ Declare @x int

يجب على المتغير ان يكون declare

ويند كذا نوع ال datatype

initial value=NULL

* هنا اربع طرق علشان نقدر ايجاد variable

static value لـ select , or set

قيمة ثابتة

→ set @x=10

* بعده نقدر select , or set

→ Select @x=100

* من هندرصنفس هنعمل ايجاد assign

@x = age from student where id=1

* من هندرصنفس هنعمل ايجاد update

update student set fname='omur', @x=age where id=9

* هنا ايجاد assign و update لا يحصل على select

→ select @x

* علشان ايجاد المتغير هنا select

assign declare run لهاها مع سطر او

: global var ②

can't declare or assign global var

* من هنقدر اعرفها او احط فيها قيمة لانها يجبر و موجودة في ال server

باتجاه الالى هنحصل في ال server من هنا رقم لعرضها بس

* بسب ابي ال و كل ار global او session بآخر statement علشانها run

→ select @@servername

* يتعروف تم ادessoar المجموع الجماع

→ select @@rowcount run last statement

* تترجع عدد ال rows ادaffected

→ select @@version

* يرجع نسخة اد sql الى مسحاع علشانها 2019

→ select @@error

* لو فـ error فـ run last statement هيطلع رفع الرسـ

→ select @@identity

* علشان اعرف آخر identity دخل منه آخر insert ولو

NULL جدول و هنا نقدر ادى insert هنجزن

→ select @x=@@rowcount run last statement

* ينفع اعد assign او global في ال

→ declare @x int = (select avg(score) from student) select, declare

Select @x

في نفس الـ صـ

→ Ex: Variable

→ declare @y int = 100
→ select $\text{@y} = \text{st_age}$ from student where id=990
→ select @y initial 100, @y modify 100
→ declare @y int = 100
→ select $\text{@y} = \text{st_age}$ from stud where address='alex'
→ select @y
→ declare @y int = 100, @name varchar(20)
→ select $\text{@y} = \text{age}$, $\text{@name} = \text{fname}$ from student where id=7
→ select @y , @name
→ declare @z int
→ update student set st_fname='ali', @z = dept_id
where st_id = 7
→ select @z

→ variable and table:
→ declare @t table (x int)
insert into @t
select st_id from student where st_address='alex'
Select * from @t
→ declare @t table (x int, y varchar(20)) 2D array table
→ dynamic top
→ declare @x int = 5
Select top(@x) *
from student

→ execute "dynamic query":
→ Select 'select * from student'
→ execute ('select * from student') query لـ execute query
وتحل الناتج بناءً على query

→ declare @col varchar(20) = '*', @tbl varchar(20) = 'student'
Select @col from @tbl XXX
→ execute ('select' + @col + 'from' + @tbl) query لـ execute query
run let's query, string let's execute

2 Control flow statement:

① if :

```
→ declare @X int  
update student set stage+=1  
select @X = @@RowCount  
if @X > 0  
begin select multi rows affected' end
```

else

begin select 'No row affected' end

② if exists if not exist:

```
→ Select * from sys.all_columns
```

نحو system schema us sys نحو

DB catalog metadata

if exist:

```
→ if exists(select name from sys.tables where name='students')
```

نحو الفرض قبلها أعل لای Create table

else

Create table students

نحو Sys.Tables دلیل name

(id int , name varchar(20))

موجود نعم انه موجود وما يعكس حاجة

نه لم يتحقق error SQL Create table

if not exist:

```
→ if not exists(select dept_id from Student where dept_id=20)
```

and not exists(select dept_id from Instructor where Dept_ID=20)

delete from department where dept_id=20

else

Select 'table has relation'

اعلم مع id relation

③ Try and catch

```
→ begin try
```

delete from department where dept_id=20

end try

وكذا ما يكتب catch

begin catch

اللى خارج catch message

select error

select ERROR_LINE(), ERROR_NUMBER(), ERROR_MESSAGE()

fun موجود

end catch

أي error will not catched

④ while :

```

m> declare @x int=10
      while @x <=20
          begin
              set @x +=1
              if @x=14
                  Continue
              if @x=16
                  break
              select @x
      end
  
```

١٥ ، ١٣ ، ١٢ ، ١١ \leftarrow معرفه هی

Condition II: $\bar{y}_t = \text{Oscillate}$ for $t \geq 14$ and $\bar{x}_t = 0$

وے skip باقی اخراجوں سے

lets loop || no break just 16 instead

⑤ Case 3

موحدون من آخر Day5

⑥ if

⑦ waitfor : Search

waitfor time:

→ execute sp_add_job @job_name = 'TestJob' [and all stored proc] ↴

begin 10:20PM هسته اساعیه sp_updatejob

waitfor time / 22:20`

execute sp_update_job @job_name = 'TestJob',
@new_name = 'UpdateJob'

end

wait for delay:

→ begin

delay sp_help بعد ساختنی

waitfor delay '02:00'

execute sp_helpdb

end

⑧ choose : Search

Ans Select order_id, status, choose(order_status, 'Pending', 'Process', 'Reject', 'Complete')
as order_status

from Sales-orders

order by order_date desc

* دو کات 1 هیئت اول کامہ لو 2 بین المآئدہ

160-

③

batch , Transaction , Script

جامعة queries معاً دلالة بعدهم ونفخ
 insert delete update run insert run
 insert run update run queries معاً دلالة
 update run error delete run

جامعة queries معاً دلالة queries معاً دلالة

create table

go

drop table

go اول اجل drop و create فصل بينهم بـ

create rule

go sp_bindrule

go اول rule او create

go اول script

لو قفتح ال DB ، على right click new query في ال DB و كل thing في ال DB

* موجود constraint او create table او script

غير data و هيكلها غير data

ار data script هو all backup

Single unit of work

جامعة queries كائنا Transaction ③

Query file (SSMS, CMD)

insert → execute (FS)

begin transaction
insert
commit

ok

Delete → FS

begin transaction
delete
rollback

error

explicit trans

begin transaction
insert
update
restart server
automatic rollback

ok

begin transaction
insert
update
delete
commit
-- rollback

لما نجي insert في execute لـ query file واعل insert صرور

لـ ldf لـ Commit لـ error لـ insert او insert ldf يكتب او insert

rollback لـ error لـ mdf لـ delete او insert ldf

rollback لـ commit او begin explicit trans او delete او update او insert

begin explicit trans او insert ldf اكمله او insert ldf

ار اى حاجة تابوها و هيستن اى ركتب

لديك execute لفروع يفتح نفس المقام خارج transaction
* update run insert ويفعلها
* قبل ما يدخل او delete او crash من ال server
Restart server
Rollback او commit او insert او update transaction
* سوال اي سيحصل ان لا يحصل restart هتروع او services لا تشتوف
لو في rollback او commit begining trans ، مفهوم update insert automatic rollback
وال insert يتضمن كلهم مع بعض وكذا ما حصل من هينق حمل او insert
دار no explicit trans او update
rollback او commit crash مفهوم في المفهوم اى اكتب
الصفرة في خط او else if explicit trans
begin try

begin transaction

insert into child values(1)

insert into child values(5)

insert into child values(3)

commit

endtry

begin catch

rollback

commit

catch try

roll back

commit

catch try

</div

dynamic query

↳ Create user-defined function like body to execute dynamic query

programmability of functions can be provided by creating user-defined functions to call them later

↳ select dbo.getsname(1) returns one value from scalar function

built-in function like COUNT, MAX, MIN etc.

② inline table func:

↳ Create function GetInst (@did int) returns table

as

return (

Select ins_name, salary*12 as totasal ونفع تقويس as return last

from Instructor where Dept_id = did

instruction of func *
as is annual Sal or

parameters, legal, createfunc func *

returns table last

) alias last salary*12 as select last

↳ select * from GetInst (20) call func

function, select last last table last col all

scalar func in schema last

③ Multistatement:

↳ Create function getstud (@format varchar(20)) Param, legal, createfunc func *

returns @t table (id int, ename varchar(20)) Jsi, returns table last

as

begin

if @format = 'first'

insert into @t

end, begin as last

select st_id, st_fname from student body

else if @format = 'last'

condition last body

insert into @t

table Jsi insert condition

select st_id, st_lname from student last select

else if @format = 'full'

insert

insert into @t

as last return Jsi

select st_fname + ' ' + st_lname from student

return select last table last insert into Jsi last insert

end

last insert Jsi last

↳ select * from getstud ('first')

→ Windowing functions

built-in functions

① Lag, Lead : row over lead والـ row over lag ←
الـ row over lag ←

→ select sname, grade, ... grade, ... sname over window

x = lag(sname) over (order by grade), first 2 col

y = lead(sname) over (order by grade)

from grades over lag, lead ←

ويعنى كذا نعمل الجدول باطريقه الـ عاليز نعمها سارى او

نـ $\text{lag}(\text{sname})$ كذا بـ lag مرتبت بالدرجات هيبيـ اسم السـ سـ لـ الـ

ار الـ row قـ قـ طـ طـ

وـ lead هيبيـ اسم السـ سـ لـ الـ row قـ قـ طـ طـ

→ select *

from (select sname, grade,

ـ كـ يـ جـ رـ فـ الـ اـ سـ مـ

x = lag(grade) over (order by grade)

وـ درـ جـ هـ الـ بـ جـ هـ

y = lead(grade) over (order by grade)

from grades) as NewTable

where sname = 'eman'

→ Select sname, grade, cname,

' x = lag(grade) over (partition cname order by grade)

y = lead(grade) over (partition cname order by grade)

from grades.

هـ نـ يـ حـ لـ اـ حـ اـ

هـ قـ سـ مـ الـ اـ دـ اـ لـ

grade يـ حـ اـ حـ اـ الـ

* لما يـ جـ هـ آـ خـ لـ مـ فـ الـ رـ اـ حـ اـ حـ اـ الـ مـ وـ قـ يـ هـ بـ

Course يـ حـ اـ حـ اـ الـ مـ صـ تـ اـ يـ بـ

② first_value, last_value:

ـ بـ هـ نـ يـ حـ اـ حـ اـ الـ جـ بـ اـ لـ

x = first_value (grade) over (order by grade)

order by grade

y = last_value (grade) over (order by grade)

order by grade

from grades

ـ خـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

ـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

ـ اـ دـ اـ لـ مـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

ـ اـ دـ اـ لـ مـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

ـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

ـ يـ حـ اـ حـ اـ اـ دـ اـ لـ

⇒ 8 - SQL , view , Index , Merge statement , Pivot tables ←

1 Index

Student

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

لو عندى Table معن PK ازاي الداتا بتاعتني
مستخزن على ال HD وفتحعامل معها ازاي ؟

- * الداتا مستخزن على هيئة data pages او packets
- * عدد ما يختلف على حسب كمية الداتا اللى مستخزن
- * او table علشان فيه غير PK كدا الداتا مستخزن as a heap
- * يعني الداتا Not sorted على ال HD

⇒ Select * from student

where Sid = 1

+ علشان كذا هيحصل Table Scan يعني هيسى

هار table مس زواهر لآخره علشان يقدر الوصول للداتا

الآن بعدين على طبقاته لو الداتا ترتيب كانت أول row خارج ال table

- * بوصول ما يليتو فيه غير PK دا هيخطا insert insert لانه sorted الداتا في الترتيب دا يليتو
- * علشان الداتا صنف sorted

لذلك لو الداتا sorted : PK , JL , HD

* الداتا هيلتون sorted على ال HD

- * وهينتج عنه حاجة اسمها clustered index وداصفيه في ال search ويسهل الوصول للداتا

* كذلك تحتاج ال insert و ال Table scan هكون أبطأ

→ الداتا هستخزن بترتيب ال PK يعني تلا أول page

خصائص 1 - 99 , وال بعد ما فيه 100 وهكذا

* هيبيتن على الترتيب بار دا PK

* كل كام Page هيلتون parent لبعض

* هيكون فيه أول دا من كام Page يعني 1, 200, 300 ...
من أول parent

* وكلهم ه يكون لهم parent يعني فيه 1, 200, 300 ...

root down

leaf level , non leaf level , root : 3 levels tree

⇒ Select * from student query حيث هو الدليل ال PK to Sid

where Sid = 804

+ علشان هو clustered index PK

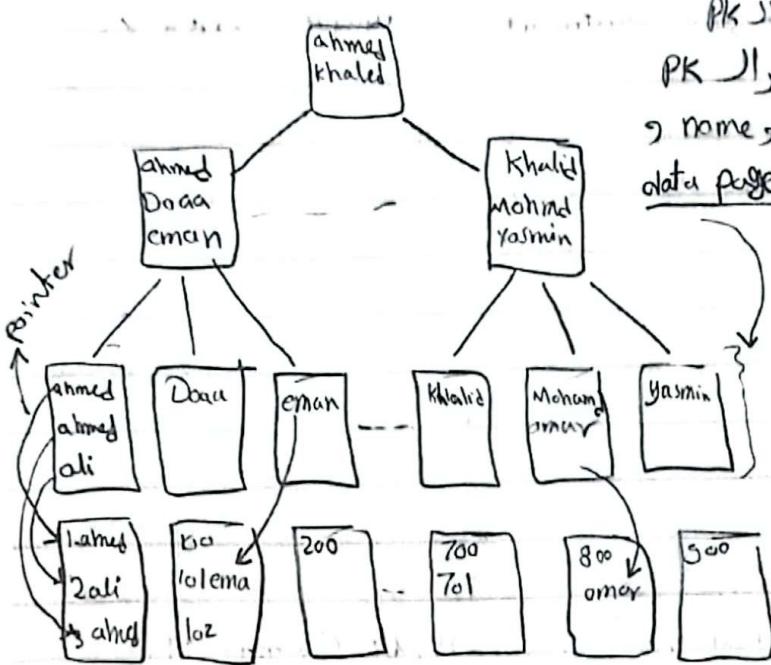
* الـ 804 هيفتح له root ينتوف له 1 < 804 < 1 يروح الملف المحدد

* 804 برضو هستوى له في الملف المحدد بس كدا الـ 700 آخر حاجة

* هيرجع بعد كدا له page 700 رابطها 804 ويقارن الـ 804 > 700 يتزلج فيها

* 900 بعد كدا 804 هيدفع إنها أصغر من 804 > 800

* **Non Clustered index** : name_pk حفظ خوارزمی search لیست ←
 * الاتصالات name_pk مخزنی name نامی و کتابخانه Table scan لیست های معرفی شده



لما نيجن نمل (index) حاجة بيرال PK
* ميأخذ no copy إلأ col اللي هو معرف
يلمدهم مع table ويلج memory إلأ عن

→ `select * from student`

where Name = 'omar'

* **PK** name دا لواړۍ همکاری کړي

* هستوف لو خا مدقق (nonclustered index) col 1 nonclustered index name

* كا هي درج لل Root يتألف من nonclus أو Tree Cusl و هما يقارن بالـ asciis

للمزيد من المعلومات يمكنكم زيارة الموقع الإلكتروني للجامعة: www.ksu.edu.sa

وَعَلَى الْمُؤْمِنِينَ أَنْ يَسْأَلُوا إِنَّمَا يَعْلَمُ بِهِمْ مَا يَعْمَلُونَ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ إِنَّمَا الْأَوْلَى لِلَّهِ وَهُوَ أَعْلَمُ

النار (H) وصيروخ (جيجي) يهاناته.

→ create clustered index myindex xxx الامر ينجز بنجاح
on Student(st_fname) الا ان PK و st_fname لا ينبع من error
table xxx has clustered index

→ create nonclustered index myindex nonclustered xxx
on Student(st_fname) indexes in file or table

→ create nonclustered myindex2 nonclustered xxx
on Student(st_lname)

Clustered index xxx على st_name كا نتاج constraint to PK

nonclustered index xxx ينتهي constraint to unique

→ create table test (id int primary key, name varchar(20), age int unique)
clustered xxx indexes in file
انما ينبع age كا nonclustered , id كا unique constraint

→ create unique index i1 unique on age
on Student(st_age) constraint unique index

nonclustered index xxx

كما ينبع i1 على st_age

عليمان كا i1 على age من حيث ان العمود

i1 index على all columns المعمول

indexxxx lege search legs

إذا أعرف اى query او query كثيرة

SPL Server Tuning Advisor & SQL Server Profiler: 2 tools

① SQL Server Profiler:

Run , save also new Trace file Profiler
select * from student where age=20 في query management

وتحل محل query to execute

stop all file Profiler كل queries
stop all file Profiler كل queries catch فقط queries engine if
engine all

٢) SQL Server Tuning Advisor: جمل queries التي هي Profiler search list

• سلسلياً من كذا ال Tuning Advisor ينبع إلى الذي نعمل عليه search list
• طائفتين من كذا ال Tuning Advisor هي Trace file Tuning التي عاليتين تشغيل ال Trace file
• و اختيار ال database المدعاة لها analysis
• بعد كذا أقل start analysis
• كذا هو في جمل recommendation index كل ما يحصل على recommendation

٣) (tempdb - msdb - model - master) by default ٤ DBs في server ←
metadata أو server configuration أو كل الـ configuration ①
* رأى تفاصيل DBs أو users أو permissions أو كل تفاصيل DBs
* كل DBs موجودة ضمن ال HD server ←
engineer Connect مثل master DB ←
create database new DB واني server ← template ← Model DB ②
هناك منها image

* يعني لو عملت DB فقط file أو views أو functions موجود عنها
* حاليات لخدمات الـ DBs موجودة في model DB
* يعني على شان كذا في طابع عاليته موجود في DB وأنها لها بقى
* نعمل على table داخل الـ Model DB
* بس دى يتغير أحياناً الجديدة بس يعني لو ضفت table أو user من الـ Model DB
* يعني الـ DB اللي له قسم مثل الـ msdb موجودة من الماول
* دى (managementstudioDB) msdb ③
* يعني أعمل backup في وقت معين أو لما تصل حاجة معاين الـ server
* يعني في jobs run query في time run ودى اسمه time → Query
* الـ msdb يعني انه يعرف دا في time بتاع الـ job جمهوار db
* دى tempdb ④

* الـ tempdb create table في temporary table مثل سوية وبيان
* الـ server أو الـ developer يستخدم الـ DB اللي عاليات creates
* من الـ runtime runtime وبيان الـ runtime
* دا هو query : ناتج الـ innerquery يتحقق الدلول في الـ tempdb كما يجيء
* دا هو query : ناتج الـ outerquery يتحقق الدلول في الـ innerquery run
* دا هو query : ناتج الـ tempdb run

لما دخل كذا اسماً DB فـ create table لـ physical table ←
drop table ومن هنفه عزيز و عزيز Physical table ←
من نوعين تاني من اب او هم local table و global table ←
: session based table) local table ①

create table #exam #tempdb table حسب # tempdb ←
(eid int,
edate date,
numofG int
) ← local table وكذا هو tempdb ←
session newquery دا في table دا اسم table ←
اوكلت newquery ايان حاولت اعمل select newquery * ←
من هنفه ←

Select * from #exam newquery اعمل table تاني بنفس الدفم في ←
والتي سن هيستخزو في ال tempdb ←
ه يعرف بفرق بين id الذي table كل دا ندك خلاف ←

هذا اقبل ال save old session او لا هنجز الرسائل ←
run time table create new table tempdb or nolocal ←
= (shared table) global table ②

create table ##exam table حسب # # * ←
(=) ← tempdb table موجود في ال ←
Select * from ##exam Shared table * ←
لـ table اصلها اعمل Select جزو session * ←
table و insert و delete و يقدر كل جزو معايри مع table ←
disconnected server) if connect table table اصلها اعمل معايير disconnecting

الى declare var table دا عمل اعمل table variable ←
ار الى run الى batch معاد ←
و Tables اربع لكتال ③ ←
batch الى var local و table variable ① ←
active session الى Connection local : local table ② ←
user الى session الى users كـ session كـ shared : global table ③ ←
drop - query بـ drop DB او create below : physical table ④ ←

= rollup ①

→ rollup and Cube:

→ Create table sales
 (ProductID int,
 SalesMan varchar(10),
 Quantity int
)

Truncate table sales

insert into sales

values (1, 'ahmed', 10), (1, 'khalid', 20),
 (2, 'ahmed', 15), (2, 'ali', 45)

→ select ProductID as X,

sum(Quantity) as "quantities"
 from sales group by ProductID

X	quant
1	210
2	230
3	90
4	260
5	790

sum & select sum & select sum & select
 هائل تانى دكتا هيعطونا مجموع كل المبيعات

→ Union all

select __, sum(quantity) from sales

→ select ProductID as X,

sum(Quantity) as "Quant"

from sales

group by rollup(ProductID)

X	Quont
1	210
2	230
3	90
4	260
5	790

X	quant
1	210
2	230
3	90
4	260
5	790

X	quant
1	210
2	230
3	90
4	260
5	790

X	quant
1	210
2	230
3	90
4	260
5	790

X	quant
1	210
2	230
3	90
4	260
5	790

→ Select productID, SalesMan, sum(Quant)

from sales

groupby productID, SalesMan

→ كـ هي جب مجموع المبيعات كل row

كل موظف هضر ضلا

→ لو علقت group by rollup
 group by rollup (productID, SalesMan)

هضم 10 rows

علشان rollup نفذ نفس funcalls sum من تانى بس معهاون Col يس المجموع

هو ProductID علشان كـ هي جب الـ sum كل ProductID كل طوف وع الـ حـ

حالـ هـ جـ بـ مـ جـ وـ هـ مـ كـ اـ هـ

لو بـ اـ 10 Proـ ID وـ اـ SalesMan هـ جـ بـ sum all Prod كل موـ ضـ فـ لـ وـ جـ دـ

:cube ②

roll up في cube يعني group by على 2 col لـ rollup
 * لو عايز أصل الناتج للبيعون يعني اجمع البيانات كل جروب
 * هم يظهر في الناتج للبيعون يعني 4 queries يعني 4 queries يعني
 * اخراج الارباح يعني اجمع البيانات كل جروب
 * من اخراج الارباح يعني اجمع البيانات كل جروب

group by Cube (SalesMan, ProdID)

SalesMan	ProdID	Quant
ahmed	1	35
ali	1	145
khalid	1	30
NULL	1	210
ahmed	2	70
ali	2	90
khalid	2	70
NULL	2	230
NULL	NULL	440
ahmed	NULL	105
ali	NULL	235
khalid	NULL	100

ناتج أول جروب
 ناتج ثان جروب
 ناتج ثالث جروب
 ناتج رابع جروب
 query
 query
 query
 query
 query
 4 queries لـ union بين
 product كل حفظ بعمل sum لـ total perf
 product JS sum
 iproducts JS sum
 كل موظف بينها sum
 كل موظف بينها sum
 وناتج كل جروب على 2 col لـ rollup

⇒ grouping sets :

⇒ Select prodID, SalesMan, sum (Quant)
 from sales

group by grouping sets (prodID, SalesMan)
 order by SalesMan

ID	Man	Quant
1	NULL	210
2	NULL	230
NULL	ahmed	105
NULL	ali	235
NULL	khalid	100

الناتج هو
 مجموع البيانات كل جروب
 مجموع البيانات كل جروب

(2) Pivot and unpivot OLAP:

rotation, groupby no col, like pivot لـ

⇒ Select * from sales

PIVOT (sum(Quant) for SalesMan

IN ([Ahmed], [Khaled], [ali]) as PVT

ID	ahmed	khalid	ali
1	35	30	145
2	70	70	90

col لـ

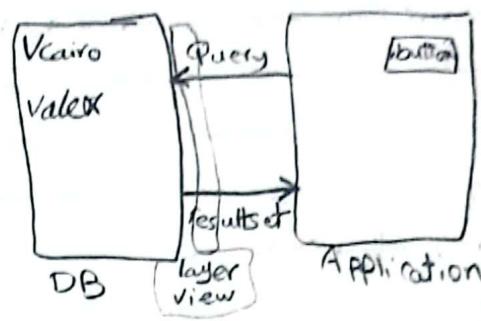
and put col JS! scroll up! كذا كانت الأشياء دفع، rotate كل 25

⇒ unpivot (search)

3) View "Standard view: data من غير view في *"

→ Create View Vcairo as
Select id, name, address
from student
where address='Cairo'

→ Create View Valex
select id, name
from student
where address='alex'



→ سيكون عند APP هلين جواه queries وال queries في متجر ال DB
Select * from student

+ لو حرض مثل button فار APP ينزل وس عليه ينفذ ال query في

* ما ذاد علىه هيروع ال Query يترى في Network وهو في ال DB يستقر وال الناتج أو اد resultset يرجع لل APP

→ أول مشكلة في الـ View الـ security : الـ query يطلع الـ data في DB الـ metadata

* لأن الـ query يطلع في الـ network و اي hacker يقدر يغير قواعد query

- خليها select or drop table

→ علشان كذا هيحصل في layer

* الـ layer دى تخلين أخف الـ metadata وجزء من الـ data لو عايز

select like, Vcairo aaw view

select statement as view

* يعترض حطينا الـ SQL و كذا جوا الـ SQL او

Vcairo Ifawl اقدر create DB

→ select * from Vcairo Vcairo بالـ select اقدر اعمل APP

* وكذا لو حصل على Vcairo الـ drop , حمل Hack حاجة هنأثر لـ ai

أولاد في الـ Vcairo و دا مني اقدر اعمل table

أولاد في الـ Vcairo كذا اقدر اعمل view

① is a select statement

① اد view عباره عن select statement

② specify user view data

② ميزته الـ كبيرة بيسعد كل dev اقدر اعمل view

③ hide DB object

→ ④ limit access of data

⑤ Simplify construction of Complex queries view

⑤ لو عندي join بيسimplify كيتر عالـ خططها في الـ view

ونجه call احتاجه عمل ما كل معه اقدر اعمل FK و AI

والكلام دا بيأثر على الـ security من اد parameters

⑥ has no parameters

⑥ اختلاف في الـ view من اد func

- ④ has no DML queries inside its body: body هو الـ DML في، وهو أي func
- ⑤ Standard view can be considered as Virtual table. كل view داتا جاهزة view
- ⑥ Insert (ie acts body هو الـ View) View هو DML أو insert أو update أو delete
- ⑦ only index view can increase performance

→ Types of Views:

Virtual table هو الـ View ①

different Servernolts view ②

→ Create view VStuds as

```
Select * from Mansoura_Server.itl1.dbo.Students
```

View is container

نقدر بـ 1000+ table

Union all

Select *

```
from SchogServer.itl2.HR.Studs
```

(search) : indexed View ③

performance فيدي طبا وصوالحبي الـ View ④

→ Create view Vjoin(Sid, Sname, did, dname) view ⑤

as

```
Select St_id, st_name, d.dept_id, dept_name
from Students inner join department d
on d.dept_id = s.dept_id
```

→ Create view Vgrades

with encryption

join على طبقات view

وكلهم في

Select sname, dname, grade

```
from Vjoin v inner join Stud_Course sc
```

on v.sid = sc.st_id

→ SP_helptext 'Vgrades' view لـ helptext view لـ helptext

create or replace view vgrades with encryption

encryption

جوازات الـ View على script لـ view

- **View 1** (one table) \rightarrow DML Jet interface \rightarrow View one table ①

مکانیزم را در طبقہ view یا table میں کاٹ واضع کرے۔

- Lname or grade must be column of table
 - در مراد Col الی محتوى فار view albums بعنوان حاجة من اینجاه
 default Value ③ allow null ② identity ①
 calculated or derived ④
 : Multi table ⑤

View \leftarrow کرنا table کی view join \rightarrow نقل مانندی delete

ار "insert" و "update" و اتصال table میکنیم ②

- Insert into vjoin insert into vjoin ایجاد error کا حل کیا جائے
- Values (21, 'nada', 700, 'Cloud') XXX 2 different table, 1 row فری
- insert into vjoin (Sid, Sname) columns ایجاد سطح ایجاد کریں
- values (21, 'nada') allows student table کو view تو یہ ممکن نہیں

With Schema binding as table schema قبل الـ table نكتب اسماء

میں اس کا degree of col → alter table

fromdbo.Instructor where dept_id=10

→ alter table instructor after column ins_name varchar(10) before ins_name
اندیشی copy از instructor کیوں کریں؟

کے ایسا اور جو احمدتِ اللہ، لارسِ آنٹافریم ایسی اور اسکے مطابق

scalar func لے جائے

index view ← لوگو

لوكست وافق ف DB no select و بـ J & B تابعه

: with check option ←

- alter view Vcairo (sid, sname, saddr) as
 Select st_id, st_fname, st_address
 from student where st_address='Cairo'
 with check option
 → insert into Vcairo
 values (3210, 'ali', 'alex')

كما الحالات ←
 مع check option ←
 بحولت اجل col → insert ←
 كنتم في هذه طاقة غير add 31،
 check جزاً لا يتحقق Cairo
 على الـ range بتامي

4 Merge

Last Transaction

id	name	myValue
1	ahmed	4000
2	ali	2000
3	omar	6000
4	eman	7000

DailyTrans

id	dname	dval
1	ahmed	5000
2	ali	1000
10	nada	3000

في الـ LastTrans والـ 2tables ←
 الجيل فيها هو البنك ، والـ DailyTrans ←
 فيه ادخالات اليومية ←
 كما المفروض نهاية كل يوم اجل ←
 DailyTrans بالـ id lastTrans ←
 للـ LastTrans بالـ id lastTrans ←

- nada و eman ، omar ← update view ←
 LastTrans 31 update 2tables الـ lastTrans الـ merge ←
 → Merge into LastTransaction as T ← update all tables if merge ←
 Using Daily Transaction as S ← update all tables if merge ←
 on T.id = S.id ← update all tables if merge ←
 when matched then ← update all tables if merge ←
 update ← update all tables if merge ←
 Set T.myValue=S.dval ← update all tables if merge ←
 when not matched then ← update all tables if merge ←
 insert ← update all tables if merge ←
 values (s.id, s.fname, s.dval) ← update all tables if merge ←

وأكتب on وجيها الشروط الـ match ←
 أو على الـ query ← update all tables if merge ←
 وجيها الشروط الـ match ←
 when matched then ← update all tables if merge ←
 وهو الذي أجل للـ lastTrans ←
 صناعي ← update all tables if merge ←
 2,1 id ← update all tables if merge ←

* وهذا when not matched then ← update all tables if merge ←
 عندما لا يوجد table Daily ← update all tables if merge ←
 Insert لـ neda إلى lastTrans ←

* وفي الآخر نصل بذلك في

when matched and s.dval > T.myVal then ← update all tables if merge ←
 لو لقينا id وجيها day مع اليوم دا أكبر منه آخر سبعة ← update all tables if merge ←

- When not matched by target then نكت المفتر و مع الجدول target not matched
- When not matched by source then by source او by target
- Merge جلب المعرفة على الجدول target *
Using الجدول target على الجدول source
- يعد الحال الى قات ما مكتشاف صفات اعتبرها by target by default ومحاتها لو في row الا target هو DailyTrans source وهو موجود في insert الى هو راى lastTrans مدخل عليه الشرط الى وهو ينبع insert و كتب عليها by source
- كما هي في اد rows الى في الجدول source ومحاته الى target insert by delete و ادخل عليه اي شرط مثل oman و omar

⇒ 9- SQL, Stored procedure, Triggers, XML Tables ←

II Stored procedure & Trigger

→ Query → Parsing → Optimize → Query Tree → Execution plan
 (syntax) (MetaData) (from, where, select)

: query ال Execution ←

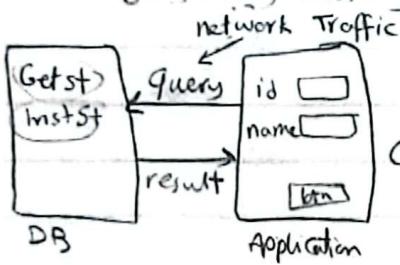
* أول حاجة بمرحلة parsing على أن يتأكد أن ال syntax صحيح

* ثانية حاجة أن تكون id student أو id متغير

* الثالث حاجة أن query tree وبيانات من يتضمن select قبل from

* رابع حاجة query tree وبيانات من يتضمن select بعد from

* الخامس حاجة execution plan وبيانات من يتضمنها



← وكل ما حصل هنا دوستنا على ال application button

وال btn داسينت أو query (Select * from student where id=4)

← لو دوستانك اد button ثانى هيمم نفس ال cycle الى

حصلت دى من الاول ودها ترجع او query

وصياد وقت لأن قم البيانات يكون كبير وصياد وقت على ان

Memory APP بين HD و APP و read و write

← برضو ال network او performance query كل ما بعد اخر دوى

أول query

← علشان كذا اختبروا ال procedure و function ال

⇒ Create procedure GetSt @id int int procedure
 كذا مثلاً باخذه حاجة procedure
 ويتضمن ال Select دى

select * from student

GetSt

↓ DB دلوتن حاجة اسمها

where id = @id

↓ select دلها مرأة ال btn

GetSt 4

بيانات فيها اهتم بالد

↓ الفوائد من ال procedure

* أول حاجة لأن علشان hacking ، كذا لوفي من هم ظاهر

إيه اللى يحصل هو ال GetSt من معروف بـ insert ، لا إيه لأن جووها

يقع بـ insert أو update أو delete أو select

باتجاتها لا بد ما يفتش في ال view و func

* ميزة تانية قدر ال characters في ال network traffic

← ابداً يدخل طابع first call لل procedure

- * شبيه بـ id بـ 4
- * يرسّ بـ query tree بـ Parsing بـ query tree بـ save
 - * نفذ ما ينفذ الـ save query tree الى المخطأ او الى صاحب المخطأ
 - * بـ execution plan
- ← وكذا يبدأ الـ call دى أي مرة تدخل call دى هي من المخطأ الى صاحب المخطأ
 - * يأخذ و يمثل بـ execution plan بـ طول و مدة التنفيذ وقت تأثير المخطأ على المفاتيح
 - ← هو الـ stored proc لو كبيرة على أقل أجزاء منها func view و call
- ← لوحدي في الـ App في 2 text boxes واحد أكتب فيه الـ id و واحد الـ name
 - * لو كتبت في الـ id = 7 او الـ name = 'ali' دوست مع الـ btn
 - ← كما الكود اللي اتبقي في الـ insert
 - ar insert into student (id, name) values (7, 'ali')
 - ← أول مشكلة عندنا الـ security wise اي تتعامل بـ direct query بـ table و cols. تجاه
 - ← مشكلة تأثر في عدد صورى كبير و أكبر في DML و حجاج
 - تأثر هنا على الـ network traffic و نفس مشكلة في الـ insert
 - ← الثالث مشكلة انه use يسأله ما هو لـ query
 - ← رابع مشكلة ان على الـ id اللي دخلته اللي هو 7 يكون موجوداً
 - عندي ويقال error
 - * الـ error اللي يظهر في الـ engine في الـ sql تأثر على الـ website او الـ app
 - ← ومفهوم الـ errors بـ nature of DML insert update delete
 - ← حل المسائل دى في الـ stored proc لـ id 7 كـ insert
 - Create procedure InstSt (@id int, @name varchar(20))
 - as
 - if not exists (select id from student where id = @id) DB engine
 - insert into student (id, name)
 - else
 - insert into student (id, name) values (@id, @name)
 - دى هي فعل الـ insert
 - كما قيم الـ id network wise
 - ← حل مشكلة الـ security
 - ← ضمن بيان الـ insert او update او insert
 - لا يزيد عن 1 character
 - لا يتغير الـ price
 - replace engine
 - replace @name, @id with @name, @id
 - الـ else و المفترض هو الـ proc تكون شامل try و catch او if و else
 - زائر 7, ali
 - دى علماً يدخل check على الـ id و دا يعني ان يحصل error لأن له الـ id موجود
 - if not exist, duplicate id

→ كذا هو ايد ال SP stored

+ بالنسبة للsecurity تخزن أسلوبيات objects زي objects func و view

+ بتاتد func parameters زي ال

+ من حزم نوال query دا بعمل call بسها في ال

+ query N execute كل مرة مفهمل query tree لـ save

+ ودا أفضل للـ networkwise ، وافضل للـ performance

else if try catch errors ياعلي الـ server

+ سهل اخذان business rules حوال

Stored procedures حوا خايل الـ DB الى سيركلية file & لـ spfile

الـ 파일 الـ SP بيتحزن فيه الـ SP اللي يجعلها

→ 3 types of SP

→ built in SP

→ SP_bind rule

→ SP_unbind rule

→ SP_help constraint

→ SP_rename

→ SP_addtype

+ يتعرض الـ object في الـ table

object لـ rename

+ datatype

→ user defined SP

→ كذا جوا خايل او SP stored procedures

select * from student

→ يقع اعلاها call بـ spfile وفي حال دا

→ GetSt (or execute GetSt) execute دا لـ spfile

→ Create proc GetbyAddress @add Varchar(20) as

select st_id, st_fname, st_address from student

where st_address = @add

Get by Address 'alex'

→ Create proc InstSt @id int, @name Varchar(20) as

begin try insert

insert into Student (st_id, st_fname)

values (@id, @name)

end try metadata

begin catch error message

select 'Error' catch, try like this metadata

end catch Error like this

InstSt 44, 'ali'

Parameter

- Create proc sumdata @x int, @y int = 100 : Calling by position
 as select @x + @y parameter كل متغيرات procedure call يدخله ←
 → sumdata 3,9 @x في المقدمة بالترتيب، و@y في المقدمة بالترتيب ←
 دلالة يعنى الترتيب مهم ←
 → sumdata @y=9, @x=4 : calling parameter by Name ←
 يكتب تمارين القيم المتغيرات بحسب ترتيب مادى ←
 → sumdata 3 دلالة لا تختلف البار 100 مرات ←
 دلالة يعنى البار 3 و@y=100 و@x=3 ←
 * ولذلك دلالة SP call اقراء القيم غيرها ←
 اى اقام وهو هجتمع او default ←
 * Insert based on execute ←

→ Create proc GetbyAge @age1 int, @age2 int دلالة SP واستخدم اوائل
 as بس اى حاجة ←

select st_id, st_name from student
 where st_age between @age1 and @age2

Insert based on select query
 بس هنا الـ SP جوال insert دلالة insert into
 based on execute ←

- declare @t table (x int, y varchar(20)) حفظ جدول table انتقام
 Insert into @t string في x و int في y دلالة 2 col no ←
 execute GetbyAge 23,28 execute دلالة @t و لازم ادخل ←
 → Select count(*) from @t دلالة اسم الـ SP وكذا هيجب كل الـ SP ←
 ، 23 و 28 ويطبع حجم دلالة @t ولو عدد ←
 دلالة select دلالة count دلالة كلاها هيطلع عددهم ←
 * return from SP ←

→ Create proc Getdata @id int as
 declare @age int

return دلالة SP من يرجع نقل ←
 غير حاجة بويها int ←

Select @age = st_age ←

* ومن ثم يتخلص return غير حاجة واحدة ←
 ما يكتفى اكتر من ←

from student where st_id = @id ←

ار proc دلالة بتاخه id و ترجعنا الـ age ←

return @age ←

ار stud الى اداره بتاخه هو دلالة ←

→ declare @x int ←

* ادار age بويها int دلالة تكون اتنا ←

set @x = execute Getdata 3 ←

عمل ادار proc دلالة كأنها ←

select @x ←

دلالة ادار proc دلالة return دلالة ←

دلالة دلالة دلالة ←

دلالة دلالة دلالة ←

دلالة دلالة دلالة ←

دلالة دلالة دلالة ←

← المفروض أن DB developer وال APP programmer يتفقون على أرقام دلائل PK التي يختارها مهندس النظم عند return لـ ما هيعرف بالحالة في الـ FK لو 200 يعني حالة في الـ FK وهذه ا

ـ علشان كده دي return ال SP مجهول علشان أرجع ارقام اللي عارفها الـ dev
ـ وار APP بيس علشان أرجع قيم دا هيفيد فار security ! الـ lوحده يرجع
ـ ساف الوجه اللي راح معندهم ايه اد behaviour اللي بيعدل عليه
ـ : output parameter

→ declare @x int ,@y varchar(20) name و ایڈج اے اے
execute Getdata 6, @x output ,@y output پڑھ لے Getdata اے
execute InsStd @x ,@y @y ,@x

→ Create proc GetMyData @age int output, @name varchar(20) output: Input output parameter
with encryption
as
begin
select @name = 'John', @age = 20
end

Select @age = st.age , @name = st.fname
 from student @
 where st.id = @age st.id = @age نتائج انتقال where @age = st.id نتائج انتقال نفس الوقت output @ info

→ declare ax int = 6 , ay varchar(20) parameters !! declare دلائل ←
 execute Get my data ax output , ay output على ال دلائل و دلائل ←
 Select ax,ay st_id input لـ input دلائل ←
 دلائل دلائل دلائل دلائل ← دلائل دلائل دلائل دلائل ←
 كـ دلائل دلائل دلائل دلائل ← Paramete دلائل دلائل دلائل دلائل ←

return * input_output * output * input *
class will work like with encryption like create a function ↪

جامعة طيبة، مشفوع بـ SP_helptext 'get my data'

→ Create proc getalldata @Col Varchar(20), @Tab Varchar(20)
as
Select @Col from @Tab لحالات ایج

execute('select' + @SQL + 'from' + @Tab) یه string چیزی که باید پر نمایند

→ `getattdata (*)`, `'Instructor'` execute `lambda` جواں کا `lambda` from

وھتوں کل ار query لے string

و كما طالعنا getalldata call للـ `getalldata` select * from Instructor.

← عوایس اور SP :

* flexibility همچنانچه query insert و select دارند اما DML دارند.

الـ security layer هو الذي يتحقق من صحة client runs all query or objects show لات تُسمى client من غير

وهي تسمى metadata لأنها تكتب من قبل hacker لخداع المتصفح

الآن نعود الى الشبكة network لـ engine ونرى أسلوب عملها

— parsing \Rightarrow الجملة \Rightarrow execution \Rightarrow cycle
business rules \Rightarrow hide \Rightarrow * parameters \Rightarrow *

DB چیزی را که ممکن است با خطا های error یا catch داشته باشد.

لیے ما بینجھنی استھم ار sp بول ال func

* علّات اد func يقع عادي أعلاها query. call جوا. query او ادخلها في براجمي

الآن دا سچب في ال SP اتن ادخلها حوا ز تانى

SP : ثالث نوع منه ال Trigger ③

Special Type of sp $\frac{9}{\text{أصل}} \leftarrow$

Call: Iglesias *
100-100

parameter λ für \tilde{w}_reg

← وَدَلَانِيَهَا إِبَارَةً as
جُوا الـServer implicit Code سُنُوق الـActions

پختہ سرور میں اکسٹریم کارڈ

كل بحثة في الـ Create Server ←
 table (Create في مستوى الـ constraints و columns و type *
 Schema (Create في مستوى func و SP و table الـ all * .
 DB (User و Schema الـ all *
 Server (logins الـ all و DB الـ all *
 Table (Triggers الـ all *
 Triggers on table ←

listen (call del or insert or update or implicit code ←
 table actions الـ all يحصل على الـ actions

delete , update , insert (i) ; actions الـ all *

log file (truncation) (listen or truncate الـ all من ي ←
 after instead of Triggers on table الـ all *

(after دل for execute) query الـ run after : after ①

→ Create trigger t1 table t1 and trigger حفظات ←
 on student draw file ip table student الـ all و كما هو

after insert t1 حفظ انج حوار triggers
 as select 'welcome to ITI' insert query الـ all *

→ Insert into Student(st_id,st_fname) welcome
 Values (77, 'ali') حوار table t1 (أو t1 حفظ ما ينفعه)

t1 all call

→ Create trigger t2
 on student

trigger t2 يجري في الوقت الـ all حصل فيه ←
 query الـ all

for update as Select getdate()

→ update Student set st_age = 1

Permission trigger الـ all كـ Instead of ②

→ Create trigger t3 on student user permission Deny ←
 instead of delete delete الـ all حفظات ←
 as select 'Not allowed for user=' + suser_name() delete الـ all حفظات ←

→ delete from Student where st_id = 779 not allowed حفظات ←

all user الـ all حفظات ←

→ Create trigger type on department
instead of insert, update, delete
as select 'not allowed' not allowed
→ department is table but
insert into table is read only
delete or update

→ update department 5000 id is illegal and not allowed →

Set dept_name='Cloud' trigger_M run
where dept_id = 5000 dept_nm error object 'is dept_name'

۱) **trigger** **call** **alpaca** **query** **ادا** **سوا** **او** **کیفیتی** **row** **او** **کد**

۲) در trigger بیان دهنده schema بنایت از object کا طور

→ Create trigger t7
on Sales. Student

+**الحالات الجديدة** after update trigger

لارزم ایجاد کردن سکمای مورد نظر با استفاده از دستور `CREATE SCHEMA` می‌تواند به صورت زیر انجام گیرد:

→ update sales.student alter trigger sales.t7

set id=4 where id=7

لہن استکریار update جو اے trigger یا func لی جو اے update اور اے اے دا جزو میں اے update اور

→ alter trigger sales.t7 if جب ار کی update ہے تو اسی کا نام دے لیں۔

on sales. Student false 'i true user func

* حنی update کیا جائے تو اسے update کی query after update as

if update(name) highly true please name will set

select 'hi' #للن و عملت set لله او آى حاجة تانى

→ update sales.student 'hi' ~~giving~~ false length

Set name = 'ahmed' where id=7

• trigger & drop , disable , enable ينفع أعمل ←

→ alter table department disable trigger ty كاً لفّر اُنْقَفِي الْعِلْمِي

→ " " " enable " to readonly writable or vice versa

وأقر أرجون بعدها بآليّة عمل enable و disable وما عالت هو كان لسعاده جوا

خاکل اور Triggers

→ drop trigger ty file line will always drop case due to