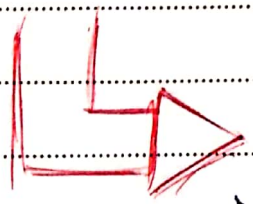
Atomicity of updates:
atomic: either success or not
يعني في اي عملية نقل او اي عملية ع بيانات ، ال database بتضمن اما العملية
تتم كاملة بنجاح ، او انها ما تتم وبترجع زي ما كانت الحالة بسلامة

concurrent: (متزامنة)
semantics: (دلالات)
constraint: (محددات)
tabular: (جدول)
i.e.: that is (id est) e.g.: for example (exempli gratia)
network model: Graph
Hierarchical model: Tree of superclasses+ children



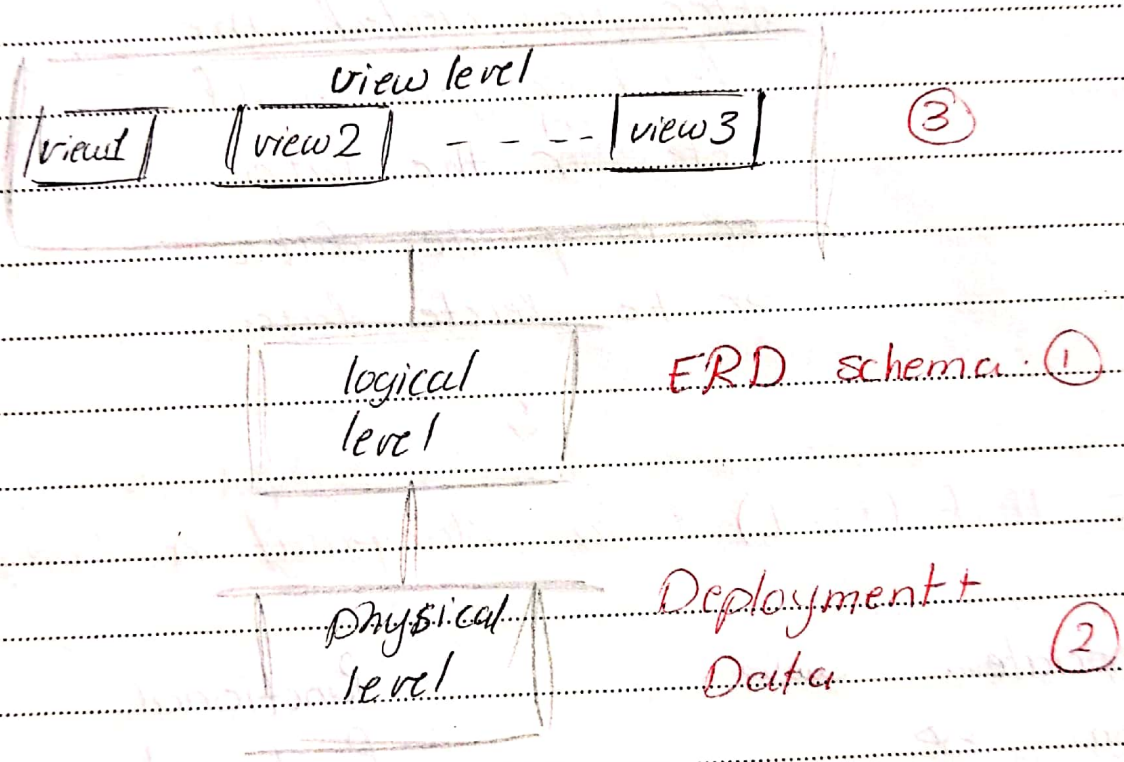
Relational Model
→ data stored in tables
→ tabular data in rows (or tuple) and (columns, or fields or attributes)
→ every table has a primary key
→ every table has a name

in slide 1.12:
between table (b) and (a) there is link from dept_name as primary key (general description) to dept_name in (a) as a foreign key (direction of linking is from foreign to primary) from (a) to (a) this case

* لما يكون عندي اكثر من table بكون اربطهن من خلال ال primary key + foreign key

Overall : عام
Analogous : مماثل
particular : خاص

View of Data



← على ورق لحد الآن احنا عملنا ال design وعملنا ال schema's ، ووصلنا لمرحلة ال deployment

↓

(DDL) Data Definition Language

↳ commands that are used to create tables of database

after you created the database you should ~~enter~~ insert the data or to update data or to delete data

↓

to do that: (DML) Data Manipulation Language

appropriate : مناسب

portion : جزء

retrieval : استرجاع

involve : يشمل

phase : مرحلة

characterize : يوصف

conceptual : مفاهيمي

Schema : مخطط

Conceptual schema : مخطط مفاهيمي

functional : وظيفي

functional requirements : متطلبات وظيفية

indicates : يشير إلى

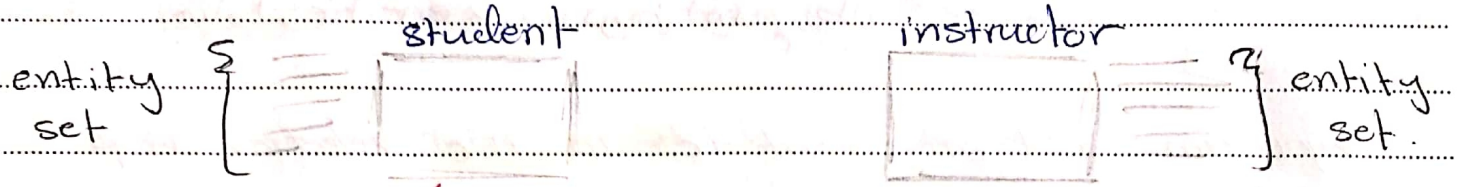
transactions : معاملات

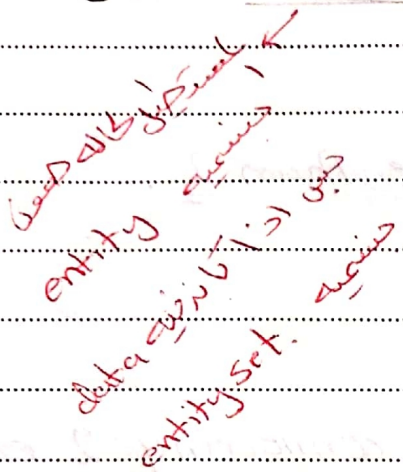
descriptive : وصفي

possessed : مملوك

Entity = Relation = table ✓
(of Entity set)

* Relationship sets





→ $\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$

e_2 belongs to E_2

there exist a set

such that

e_1 belongs to E_1

and that for example

E_1 → student

E_2 → instructor

Example:- (44553, 22222) ∈ advisor

→ the relationship between them.

student number e_1

instructor number e_2

associations : روابط
pictorially : بالصور
the relationship set → هو الخطوط اللي بتوصل بين جدولين
مثلًا 2 entities : وبربطوا بينهم بعلاقة

→ you can add an attribute to the relationship.

* Degree of a relationship:
could be binary ~~[illegible]~~ or more than 2.
binary, less, or higher.

→ binary relationship: one relationship connecting 2 entities.

Attributes representation in ER Diagram :-

in table:
- components → shifted
- multi-valued → { . . . }
- attribute followed by function sign → ( )
  derived

in diagram:
- components → are like children leaves in a tree.
- multivalued → two circles within each other
- derived → dotted line

why : لأنه بس اعمل ديزاين الـ table بحتاج اطلع الـ attributes
بعمل ديزاين الـ table
على شكل خصوصًا بس هو مش اساسي من اجزاء الـ table

([illegible] attribute، بيسمي)

← لما يكون في entity ، وأما attribute ، وهاد ال attribute عبارة عن واحد من مجموعة ال components لل entity ، وهو نفسه هاد ال component إله كمان components ، فهيك بنسميه فقط composite مش component ، لأنه هو كاسم attribute ع كل ويتم بنتعامل مع ال components تبعونه مباشرة.

← دائماً ال components اللي بالآخر هدول بكونوا اسمهم leaves في ال components ، ال composite attribute اللي حكينا عنه مؤخر

← غالباً ما بلاقي attribute → derived / composite ← بنفس الوقت

Primary key for Entity Sets topic :-

① Super key? → primary key could be composite, → containing more than one attribute connected to each other, or simply one attribute.

(primary key ← ابتداءً من ال ... مع اي شيء آخر) (بتكون عبارة عن ال super key)

② Candidate key

↳ minimal super key! ال super key اللي فيها أقل عدد attributes

→ several candidate keys may exist, one of them is selected to be the primary key.

← لو كان في أكثر من single candidate key ، بنختار منهم اللي بكون فيه أرقام ، لأنه دائماً ال query بين بتتعامل مع numbers أسرع مما بتتعامل مع characters ← better performance

واختار دائماً الأشياء اللي you can control ✓ ، ولو كانوا ثنين ، بنختار اللي أقرب للمجال اللي أنا بصمم ال database عشانه ، مثلاً جامعة بختار ال ID ، ما بختار مثلاً الهوية

Example:-

student_ID	S_Name	D_Name	Entities
1	Ali	CS	e_1
2	Suha	CIS	e_2
3	Salem	CS	e_3
4	Alia	IT	e_4
5	Ali	CS	e_5

superkey = { {ID}, {ID, Name}, {ID, D_Name}, {ID, S_Name, D_Name} }

لـ أي اشي مع ال primary key بيطلع super key

candidate key = {ID}

لو كان عنا ال candidate بيكونوا {ID} ، {S.D name} ، بنختار ال ID
كـ primary لأنه رقم ، ولأنه we can control it ، اسماء الناس اشي احنا ما النا
عليه control

distinct: مميز
perspective: منظور
suffice: تكون كافي
cardinality: العلاقة العددية

Subject : / /

choice of primary key for Binary relationship sets

when you have either one-to-Many or Many-to-one relationship, such as:

Department table (1) → Instructor table (Many)

In Mapping we take primary key of department table, which is (one) side, heading to (Many) side as a foreign key in there

So depending on that i primary key of (one) side, may have the same value repeated for more than one instructor, then primary key of a relationship set in such cardinality, is the primary key of (one) side

Important note:-

Degree : relationship is Binary , less , or higher

cardinality: " " one-to-one , or one-to-many , etc.

* (many) side, means one or more, which means it does not always have to be many.

it may include (0)

Constraint: عائق / قيد

associated: مرتبط بشيء ما

→ we say one-to-many or many-to-one reading from left to right.

Total and partial participation:-

total

↓

كل e لازم تكون عاملة علاقة وحدة على الأقل مع الطرف الآخر

partial

↓

ممكن تكون في e مش عاملة اي علاقة مع ال entity set المقابلة

↓

it may possibly include (0)

اللي لو هي many side ممنوع تكون possibly including(0).

Again :-

- Degree : such as binary.
- Cardinality : such as one to one
- participation : total / partial
- constraints things you wanna represent in diagrams for a specific concept.

in slide 35 / chapter 2 :-

→ section entity set does not have any primary key, so its participation is total, because it should have an entity from identifying entity set, to distinguish it.

→ doubled diamond is about weak entity set and identifying relationships.

→ doubled square is about weak entity set.

→ No primary key + total participation, doubled diamond + square : weak entity set + identifying relationship + discriminator's

▶ Subject : .... / /

Chapter 3 / Mapping

you go this way:-

ERD → Relational / Schema → Design coding

↙

ER-diagram.

The Bold line:-

Map ERD → to → Relational

Relation:- علاقة

① Map Regular Entity

- ال multi-valued attribute بينفصل في table خاصة، لكن بنحافظ على ربطه بال table الأصلي، باستخدام ال FK وال PK

- ال PK تبعه رح يكون composite، مكونة من ال PK تبع ال table الأصلي، واسم ال attribute نفسه، مع بعضها PK، لكن ال PK تبعت ال table الأصلي حالها تعتبر FK داخل ال table الجديد

وهاد كله لأنه ال table الجديد جزء من الأصلي، لازم يضلوا مربوطين ببعض، وبسميها جاية من FK لـ PK

Subject : .... / /

② Map weak entities:-

- هراح نحط ال Fk تبع ال strong entity مع ال weak entity علاك
Fk ، ويكون Pk مع ال discriminator

- بنربط بين ال strong وال weak عن طريق ال Pk و Fk
جالي من ال Fk في ال weak ، ودا هيبقى ال Pk في ال strong.

③ Mapping Binary relationships

* One_to_one relationship

بنهتم بالامور التالية :-
- مين الي ضروري يكون الصيغة ال relationship وبيحط ال Pk تبع ال entity التانية عنده بالاسم المتناسق مع صيغة ال relation الي بتعكس اوضيه

- مين الي اذا اضفته عند التاني بحقق performance وبسرعة اعلى.

mandatory : إلزامي

Mapping summary :

1. Map regular entities ✓
2. Map weak entities ✓
3. Map Binary Relationships
   - One-to-many
   - Many-to-Many
   - One-to-One

One-to-Many

- creating relations normally for both sides
- PK of 1side included as a foreign key in * side
- relationship attributes included in * side

Many-to-Many

- creating relations normally for both sides
- creat new relation deriving its name from the relationship & including PK of each side in as FK's to be composite PK of new Relation.
- new relation has the attributes of relationship

One-to-One

- Same here.

mandatory to optional.

- include PK of one side, as a FK in the other side (considering mandatory + performance)
- include relationship attribu to the same relati

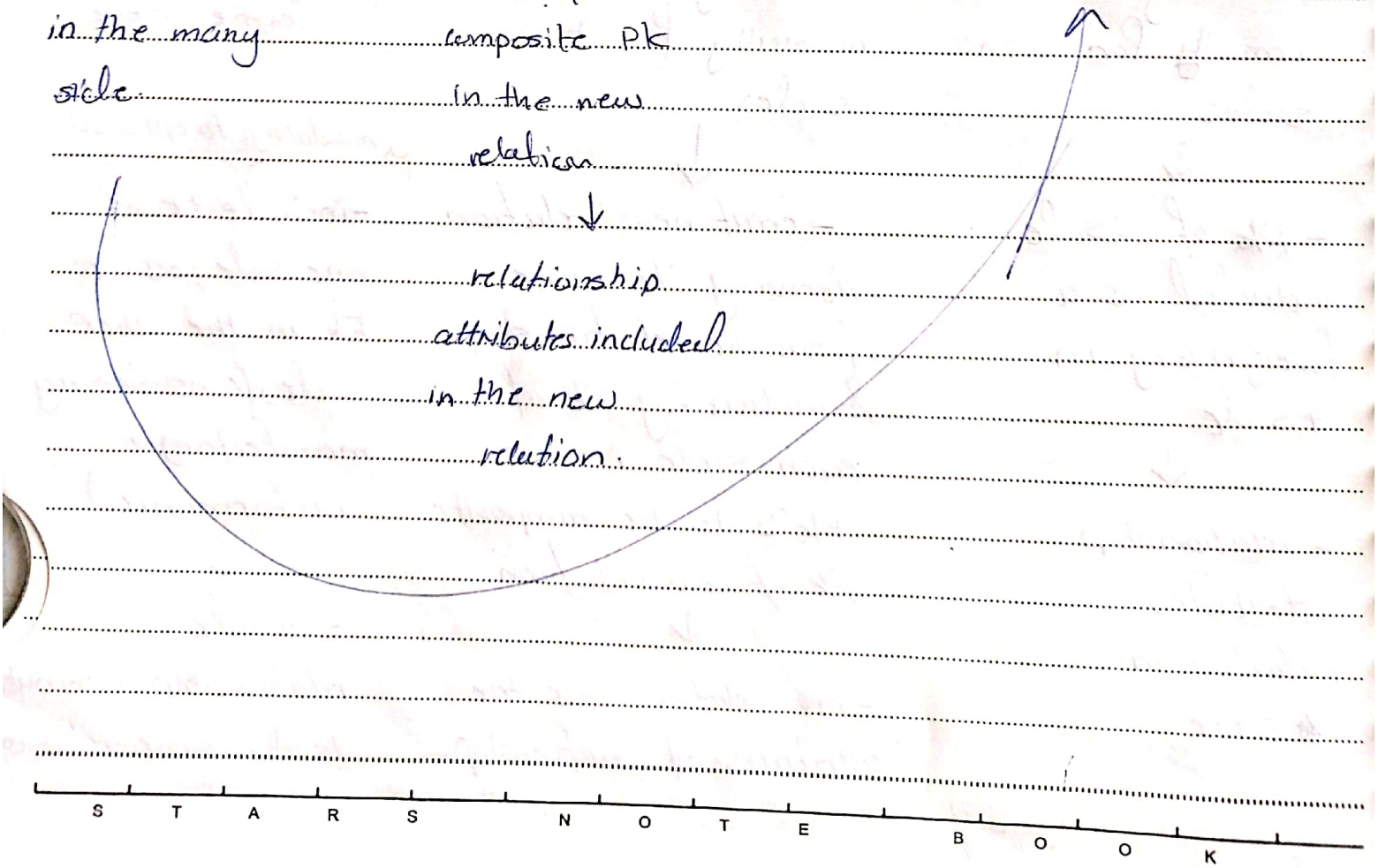
▸ Subject : .....

(4) Map Many Relationships

one to many → create a relation for entity → include Pk of the entity as Fk in the same relation with different name → relationship attributes included in the many side

many to many → Same here → create new relation and include Pk of the entity as Fk twice → composite Pk in the new relation → relationship attributes included in the new relation.

one to one → Same as one to many



Subject : .... / /

⑤ EER Mapping: Enhanced Entity Relationship

Enhanced → معزز

← لما العلاقة تكون disjoint ، ال General attributes بنزلهم لكل ال specialized tables

← لما ماكانت disjoint ، يعني حتى overlapping ← ال tables بجيبلها ، ما بنزلهم ال general attributes

* Normalization *

بيكون عندي huge table زي ال excel file ، مطلوب مني نحول هاد ال table لكل في database design ← حتى يكون سريعا اعلى

Optimizing: تحسين
accomplished: منجز
integrity: سلامة

*Goal:- avoid data redundancy.

*أي tables ، بنفصلهم عن بعض ، لا بد يضلوا موجودين ببعض

* functional dependency

3 cases:-
- 1 defines 1 ✓ always valid
- M define 1 ✓ valid for dependency
- 1 defines M × not valid for dependency

between attributes ←

example:-

ID	Name
1	Ali
2	Ahmad
3	Khaled
4	Ali

1 ← 1 and M ← 1

← ال طالب ID الو اسم

← عندي تشابه بالأسماء ، جميعًا 4,1 Ali لأني بتعامل مع data

M → 1 ← طبيعي

This case is valid such that many define 1

لأن عندي تشابه بالأسماء ، قدرت أميزه بين عن طريقة ال Many ID's

Subject : .... / /

Tricky :-

ID	Name
1	Ali
1	Ali
2	Ahmad

→ مع ال (1) تكرر هون بس
لس حسب ال data بتعتمد
1 → 1 → Redundant Data
1 → M مش

Clerk :- موظف
transitive : مشتركة

Normalization example 1:- (Case study slide 9)
in step 2

ItemNo ⟷ Description (one-to-one) (valid ✓)

← كان تكلم خاص ال UnitPrice مع ال description وال item
في ال table مع بعض مساري، والأفضل هيك، على اعتبار
ال logic لا يكمل.

Slide

Scanned with CamScanner