

"Data Base 2021 الميزه"

Definitions:

- 1. entity → is a thing That
 1. must be exist
 2. can be describe
 3. distinguishal: from other objects.

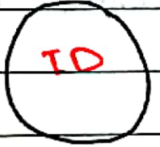
هو شيء موجود لدينا، له صفات وتغير، أم صفة واحدة عن الـ objects
 * مثلاً الكمبيوتر هو شيء له صفات وتغير، اميزه عن الكرسي، هكذا.
 نبرسمه على شكل مستطيل

En - name

2. Attributes:- The characteristics that distinguish the entity, and have Types.

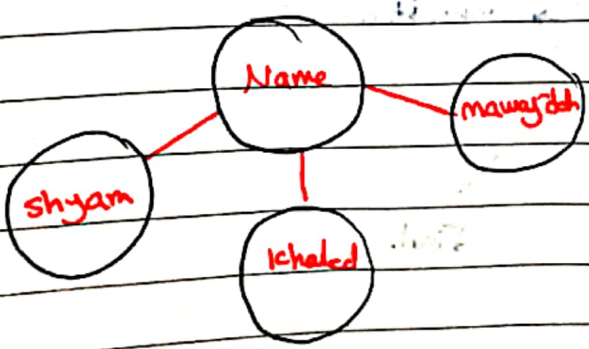
الصفات والمميزات التي تميز الـ entity، لها انواع :-

- 1. Simple :- Can not be decomposed to smaller Attributes
- الحقل لا يمكن تجزئته الى اجزاء اصغر مثل :- الرقم الوطني، الرقم الجامعي، ويسمى :-



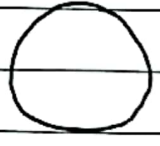
2. Composite :- Can be composed to smaller Attributes.

الحقل يمكن تجزئته الى اجزاء اصغر مثل الاسم، مكانه الاقامه، ويسمى :-



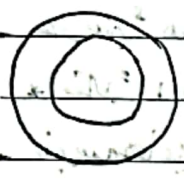
3. Single Value :- Cannot Take more Than one value

سبب ال Simple من القاد هو انه الحقل ما يوز اكثر من قيمة واحدة ويسمى :-
مثل ID



4. Multi-Value :- Can take more Than one value.

الحقل يأخذ اكثر من قيمة مثل رقم التلفون و double shape
مثال :-



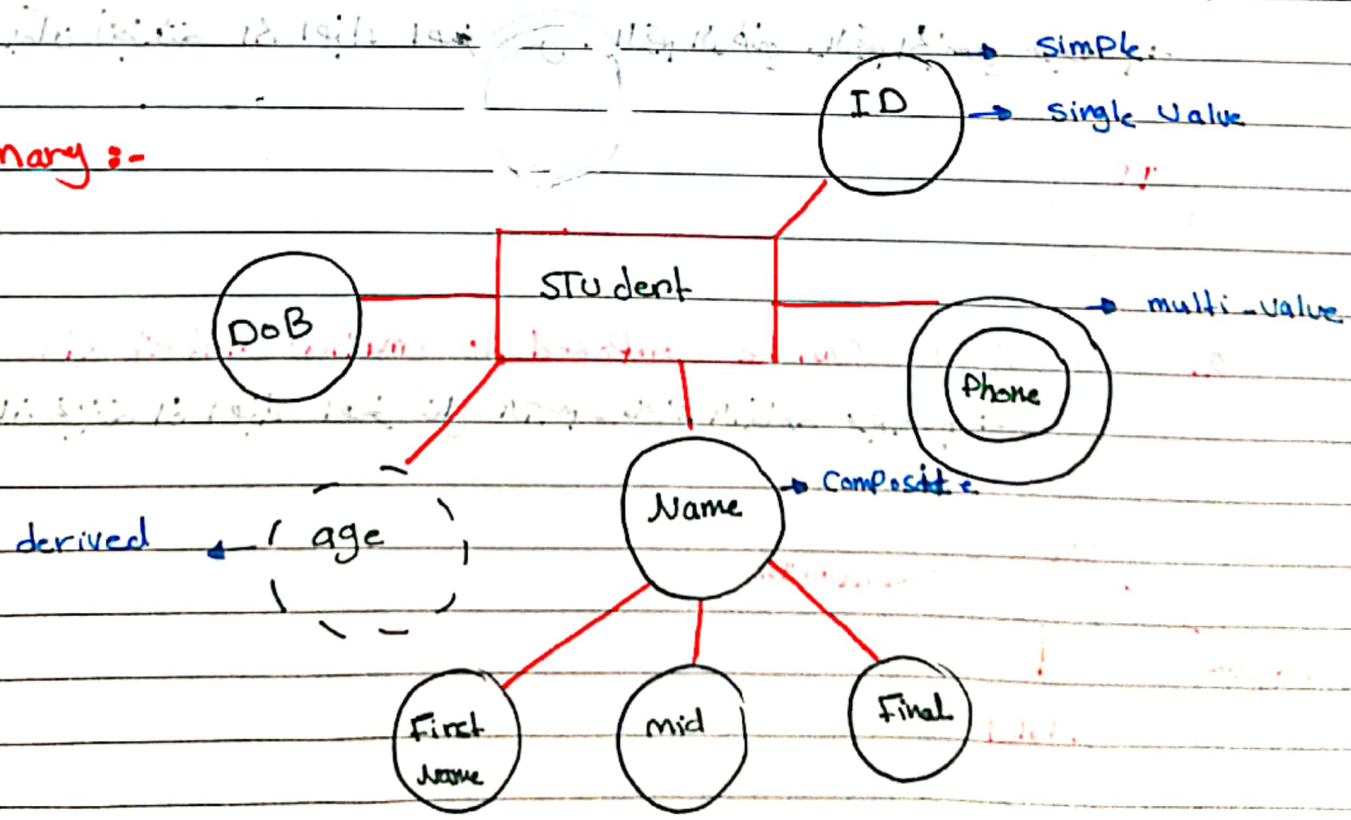
5. derived :- For Attributes That can be Calculated or executed

From other Attributes.

هو سبب انه لا يوجد له وجود مستقل في الجدول بل هو مشتق من
البيانات الموجودة في الجدول
مثال :-



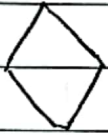
Summary :-



- Note :-**
1. 2- entity with same name Not Allowed
 2. 2- Attributes with same name in the entity Not Allowed, but in different entities are Allowed.

علاقة

- Relationship :- an association between two or more entities.
 :- **علاقة** 'only' between entities. **Attributes** **علاقة** **علاقة**



- Properties of Relationship :-

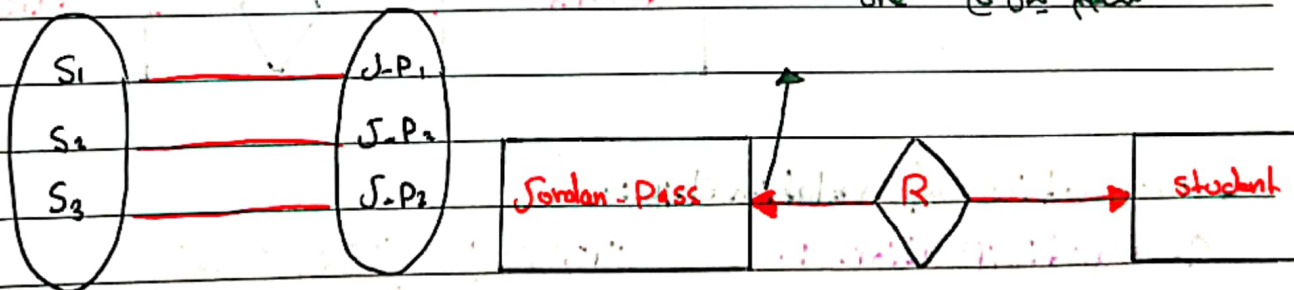
Cardinality :- **علاقة** **علاقة**

- (A) one to one
- (B) one to many → many to one (many)
- (C) many to many

- one :- have directed line
- many :- don't have directed line

Note :- one :- have zero or more 0..1
 many :- have one or more 0..1

(A) one to one

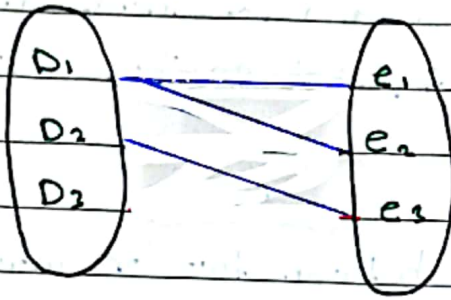


Student **Jordan-Pass**

- The student has one Jordan Pass Port.

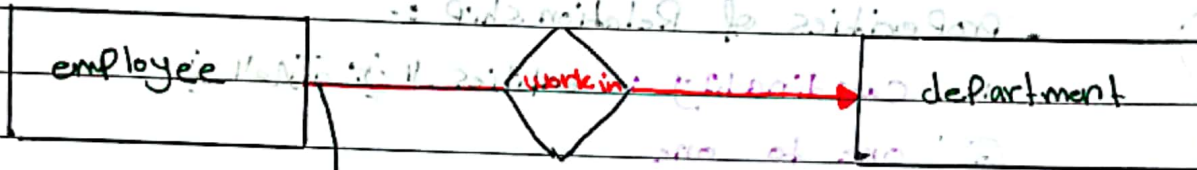
الطالب له جواز سفر واحد، و الجواز يحمل اسم طالب واحد.

(B) one to many :-



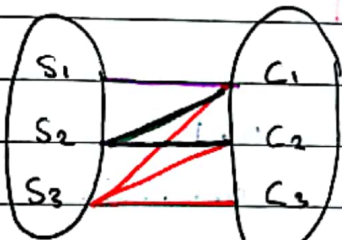
The department has more than one employee and the employee works in one department.

Department employee



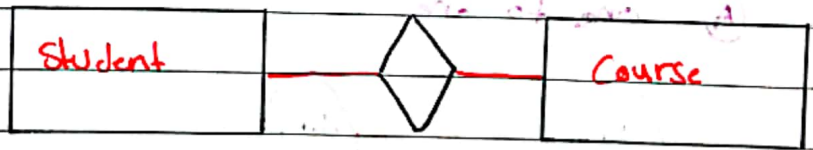
many

(C) many to many



Student Course

There are many student in many courses.



Student

Course

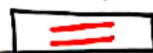
Properties of relationship :-

2. Participation :-

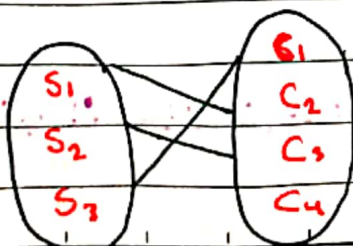
يقدم على المشاركين في العلاقة

(A) Total :-

جميع الموجودين في العلاقة مشاركون فيها



مشاركة جميع الموجودين بين ال entities



Student

chair

كل الطلبة قادمين في كراسي نس متى كل الكراسي قادم عليها

Total من جهة ال Student

(B) Partial :- على الأقل هناك واحد من المجهولين غير متساو في العلاقة
Partial من جهة الكراسي



- كل طالب له كرسي واحد وكل كرسي يقع عليه طالب واحد.
العلاقة بين الطالب والكرسي هي علاقة جزئية.

Properties of relation ship :-

3. Degree :- تعتمد على عدد ال entities في العلاقة

(A) Binary :- between two entities



شخصيا (خوفه كراهه) 😊

(B) unary :- have one entity in The relationship

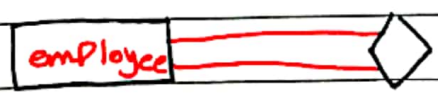
ركن يعني - 😞

ما هي العلاقة خاصة، ليس؟! هذا احيا كليا انه العلاقة (relation ship) تكون بين
Two entities صوع! هذا هو العلاقة تكون بين ال entity (قسم) ، كيف؟
مثلا كذا الموظف ورئيس القسم ، مغليا هم الاثنين موظفين فالتالي العلاقة تكون بين
موظف وموظف



Note... Page! 3 : ليس هذا الشكل فقط، ممنوع، ذلك 😊

لذلك نرسمها بهذا الشكل



(C) Ternary :- between three entities

* بهذا اول زمين فقط

(D) Quadry :- between four entities

Properties of relationship

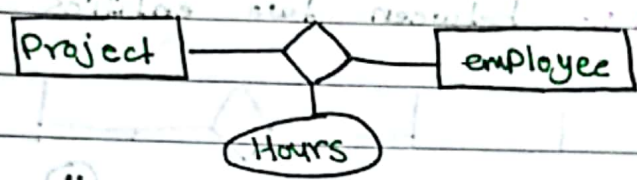
④ Relationship Attribute :- describe the relationship between entities

علاقة (Relationship) بين كيانين (Entities) لها خاصية (Attribute) خاصة بها، تسمى علاقة الكيان (Entity Relationship Attribute).

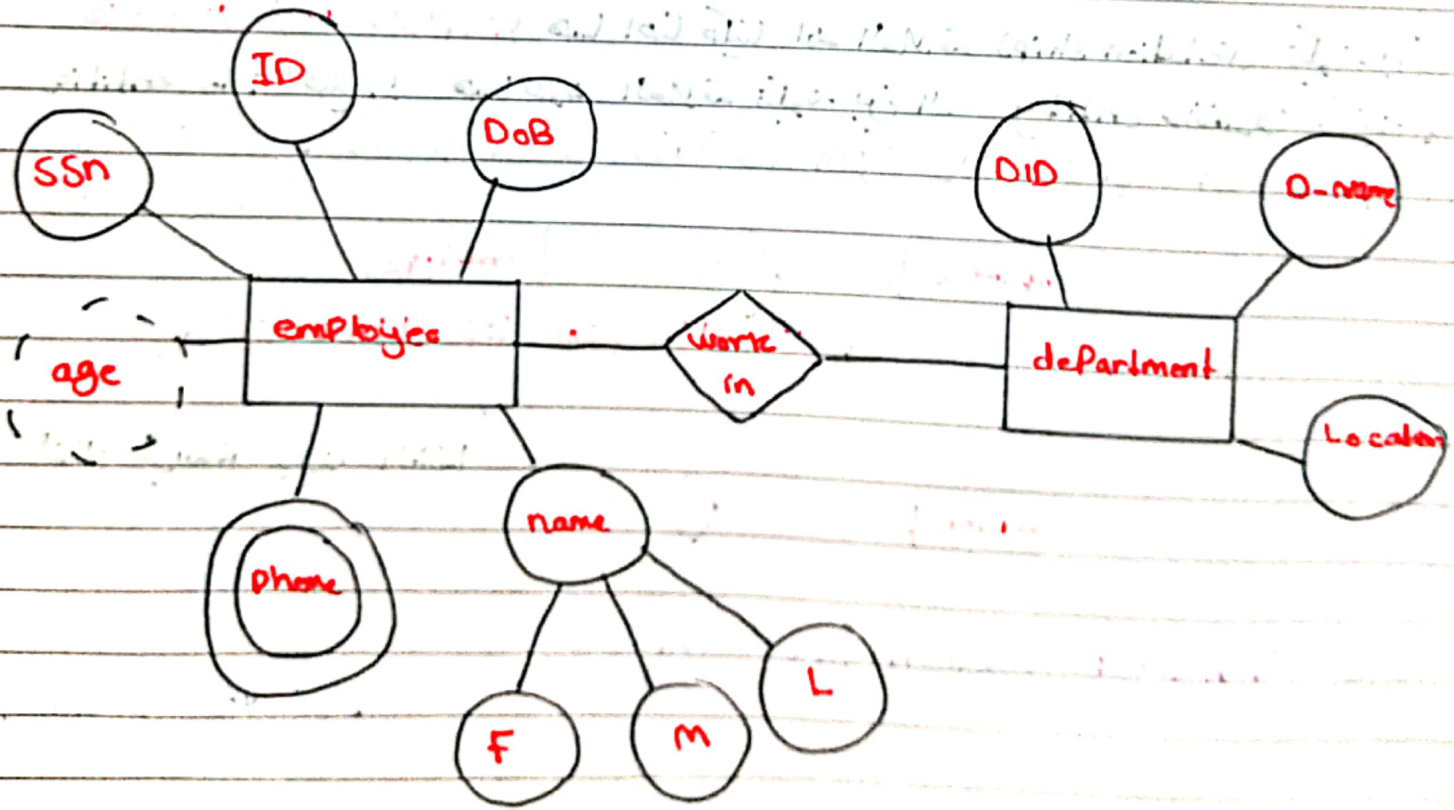
مثال: Employee و Project، العلاقة بينهما هي Hours.

علاقة Employee و Project هي hours. علاقة Project و Project هي Project.

علاقة (Relationship Attribute) بين كيانين (Entities) هي Relation Attribute.



Summary :-



Logical design (Mapping ERD)

تحويل كل Table الى entity

Relationship الى relation

relationship :- **Table** = relation

- Schema = Table = relation

- The entity name is the same of schema name.

- entity name is the same of schema name.

Mapping ال Schema

Mapping For (one to many) :-

1. Schema (one) الى Attribute

2. Schema (many) الى Attribute

Primary key الى one Foreign key الى many

Primary key الى one Foreign key الى many

Mapping For (many to many) :-

1. Schema (many) الى Attribute

2. Schema (many) الى Attribute

Two entity relationship الى PK new schema

Mapping For (unary) :-

Schema (unary) الى Attribute

PK الى Attribute

Mapping For (relationship Attribute)

Relationship Attribute الى Schema

FK الى one to many

Note:- mapping الى Composite

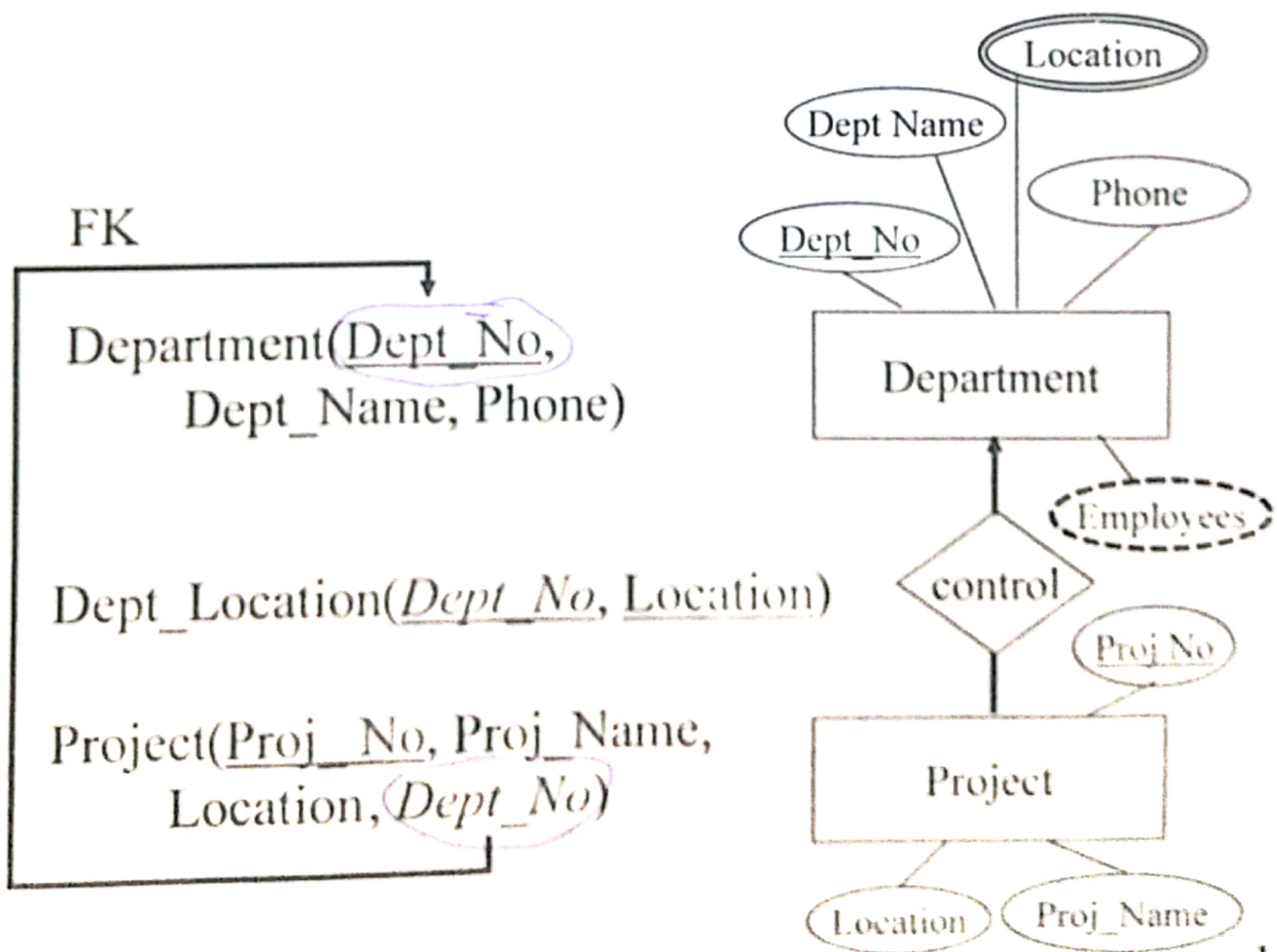
Map Binary One-to-Many Relationships

Create a relation for the two entity types participating in the relationships (step 1)

include PK of the entity in the one-side of the relationship as a foreign key in the relation of the many side of the relationship

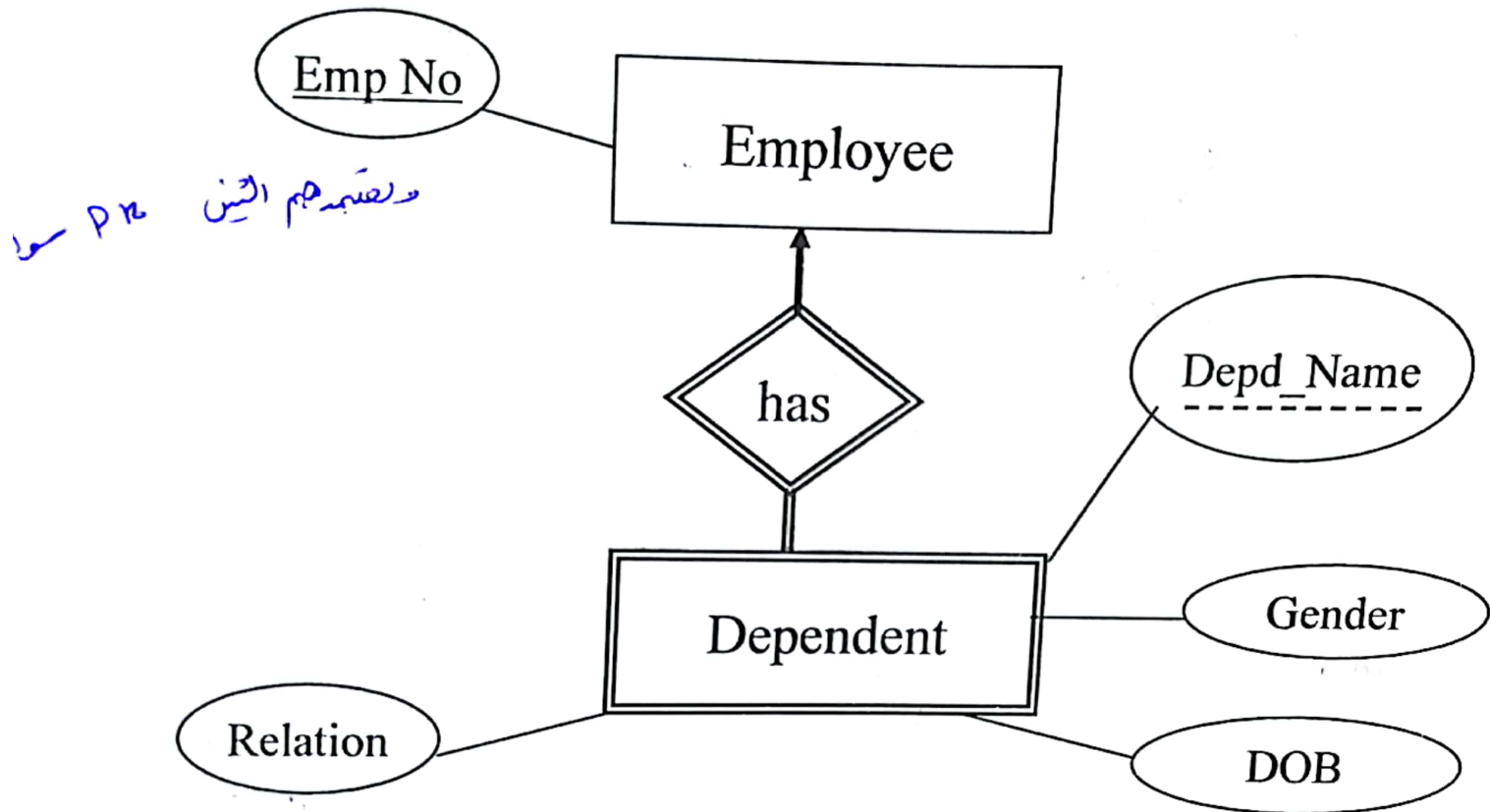
include any attributes of the relationship to the relation of the many side

9



10

one to many relation
 dis Pr & Emp & P.K. relation



و تصبیہم التین
 Pr صحت

Dependent(Emp No, Depd Name, Gender, DOB, Relation)

FK
 Employee (Emp No,)

Mapping for strong entity to weak entity
 و ربطی ال schema تانی
 ال weak و يكون كذا

Two - Pr
 From employee From dept (weak)

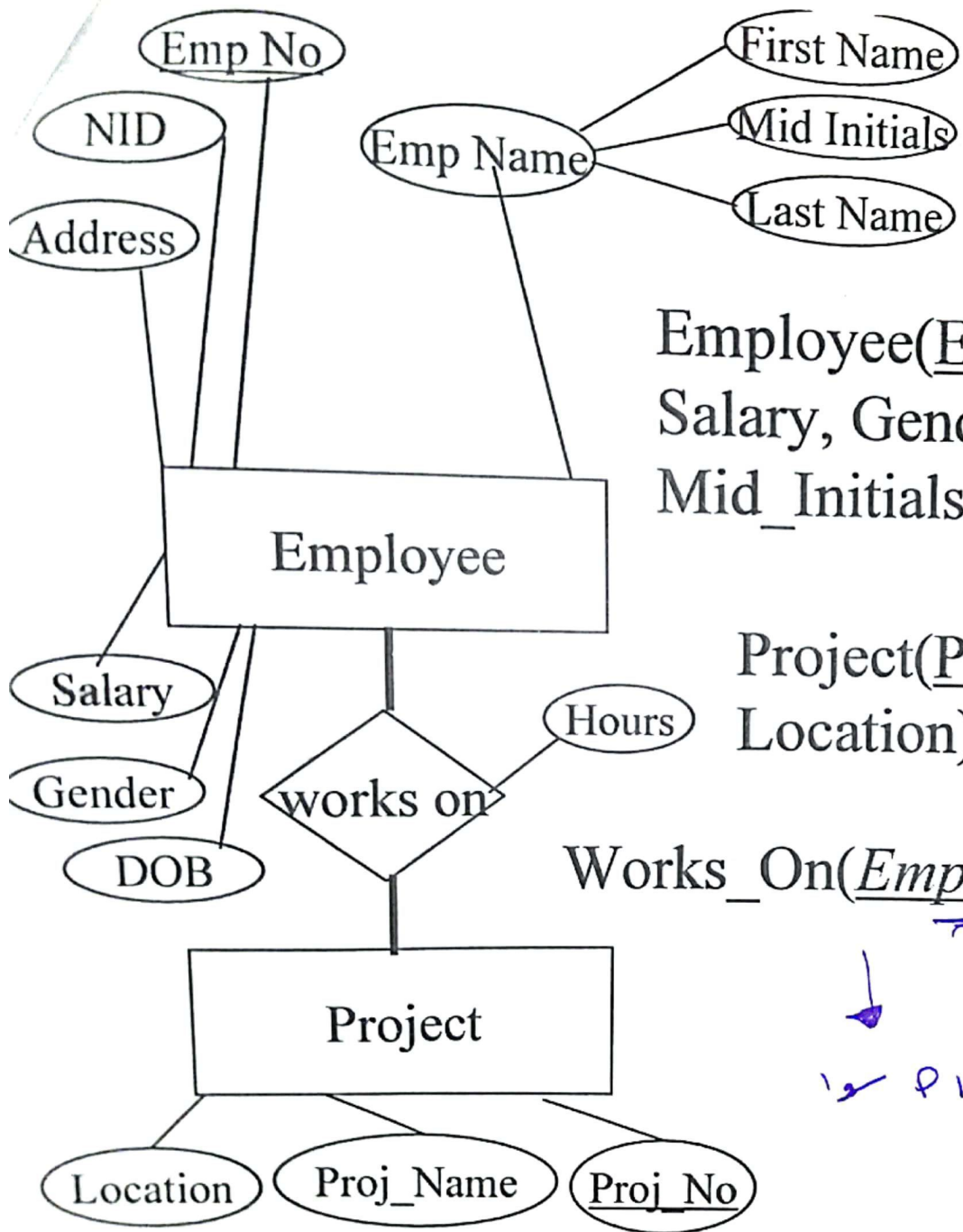
Map Binary Many-to-Many Relationships

Create a relation for the two entity types participating in the relationships (step 1)

new schema
نسبوي
رابطه فدها
P.R من
الجهتين

Create new relation and include PK of each of the two participating entity types as FK. These attributes become the PK (composite)

include any attributes of the relationship to the new relation



Employee(Emp_No, NID, Address, Salary, Gender, DOB, First_Name, Mid_Initials, Last_Name)

Project(Proj_No, Proj_Name, Location)

Works_On(Emp_No, Proj_No, Hours)

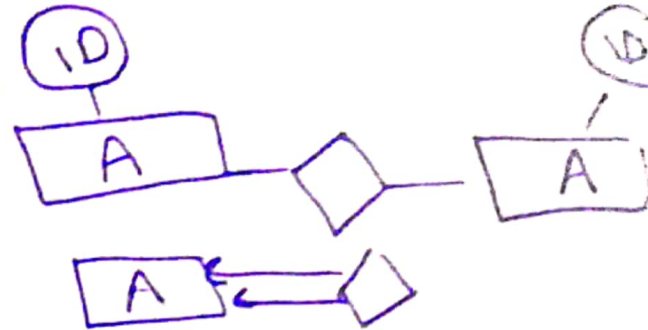
FK

FK

التنين مع يتكلموا الي ال P 12 سوا

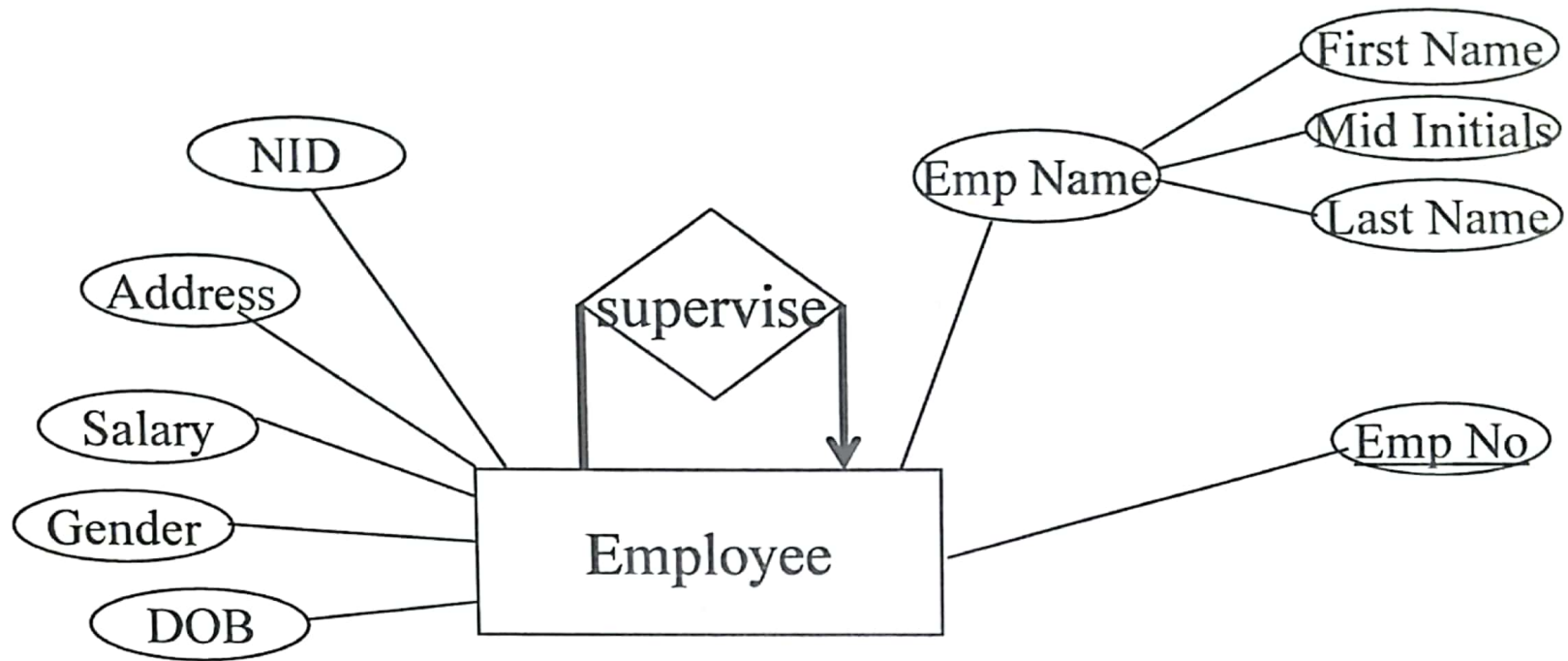
4. Map Unary Relationships

هذا السؤال صنف (خفقا للتبسيط)



Procedure depends on both the degree of the binary relationships and the cardinalities of the relationships

يُنقَل الـ PR من و هو الى التاليه بس لازم اعني اسمه
لانها حابصير يكونها نفس الاصلي



Employee(Emp_No, NID, Address, Salary, Gender, DOB, First_Name, Mid_Initials, Last_Namet_No, Supervisor)

FK

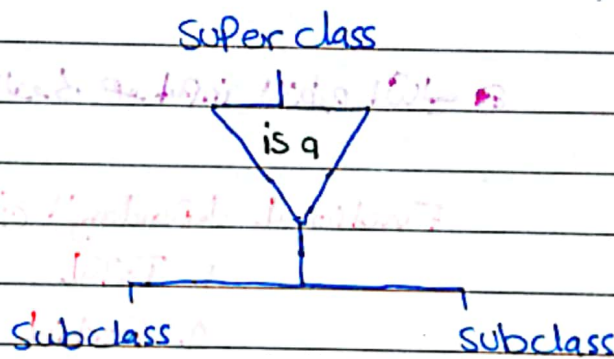
نفس (Emp-No) ابی
معیر اسامی

- important Note:-

1. derived entity mapping لا ياتي شكل
2. multi-value لا ياتي schema entity لا ياتي
3. PK لا ياتي entity multi-value

Extended Entity relationship model (EERM)

هو نوع من انواع entity و هو نظام قواعد البيانات inheritance (الوراثة)
Super class entity و Attribute entity لا ياتي
Sub class (يرث entity) و يرسم :-



Sub class لا ياتي PK و Super class لا ياتي Primary key

Type for inheritance :-

1. Total and disjoint
2. Total and overlap
3. Partial and disjoint
4. Partial and overlap

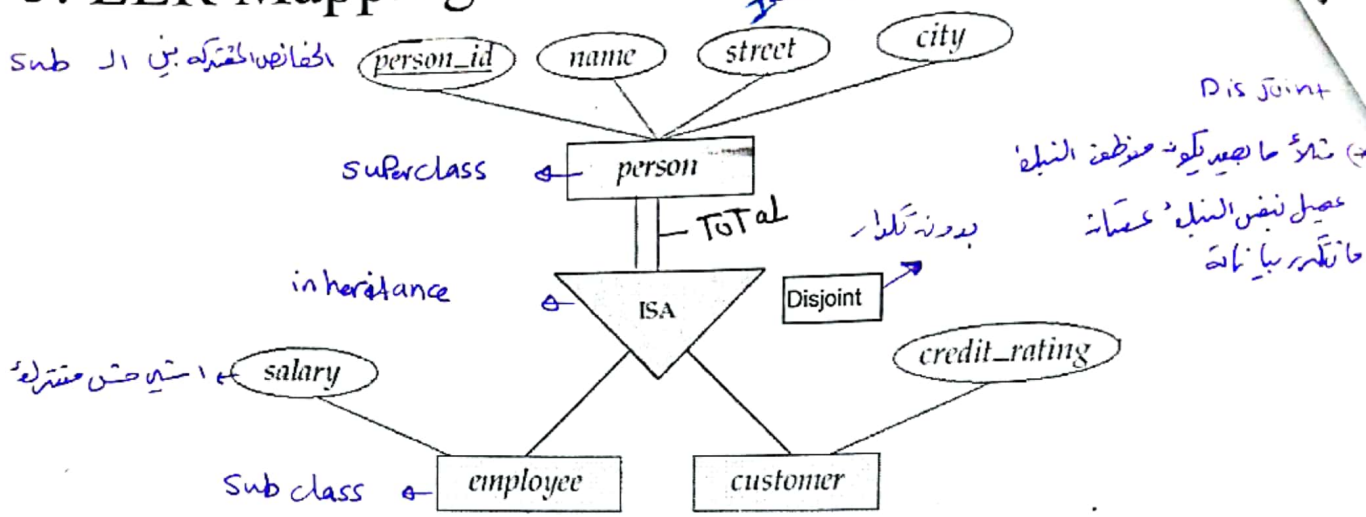
Two schema disjoint :- لا يسمح تداخل البيانات و هو انه لا (D)

3. schema overlap :- يسمح تداخل البيانات

3- " total :- لا يسمح تداخل البيانات و فقدها

3- " partial :- يسمح تداخل البيانات و فقدها

5. EER Mapping: Case#1 (Total and Disjoint)



We need to create 2 tables:

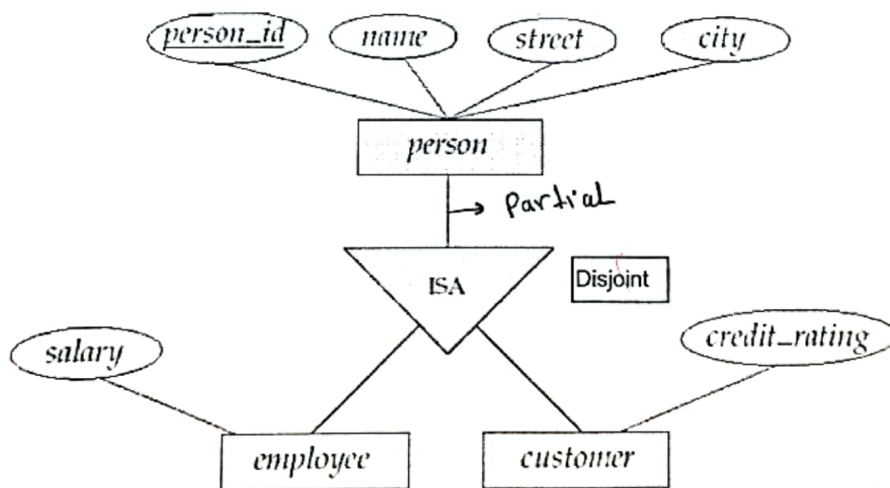
employee = (Person id, name, street, city, salary)

Customer = (Person id, name, street, city, credit_rating)

As it is total, no need to create person table

As it is disjoint, there is no redundancy as no employee is customer and no customer is employee.

5. EER Mapping: Case#2 (Partial and Disjoint)



We need to create 3 tables:

Person = (Person id, name, street, city)

employee = (Person id, name, street, city, salary)

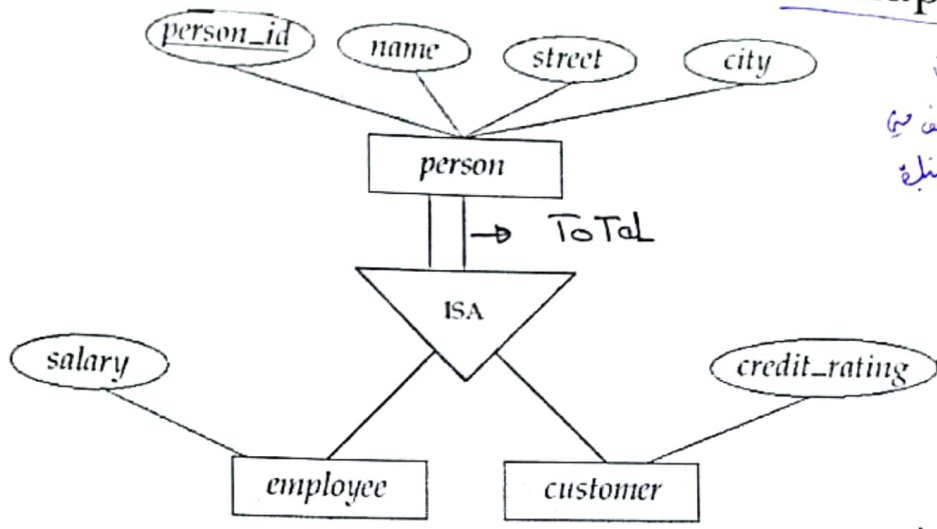
Customer = (Person id, name, street, city, credit_rating)

As it is partial, we have to use person table.

As it is disjoint, there is no redundancy as no employee is customer and no customer is employee.

Qunt

5. EER Mapping: Case#3 (Total and Overlapping)



مجموعه تسمى البيانات
 مثلا سكن بلون المرفق في
 البنية هو مرفق نغص البنية

We need to create 3 tables:

كرواها

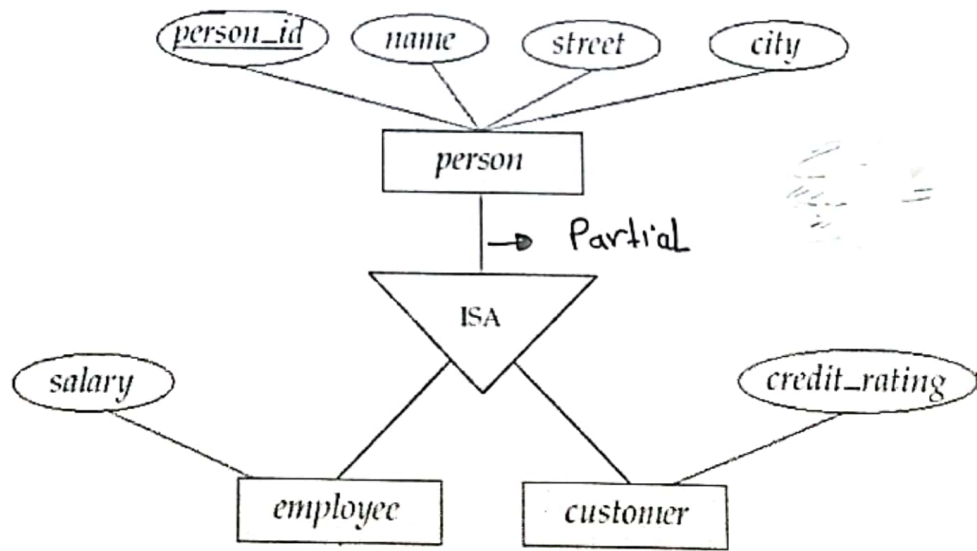
Person = (Person id, name, street, city)

employee = (Person id, salary)

Customer = (Person id, credit_rating)

Although it is total, but we still need to create person table because otherwise, there will be some redundancy.

5. EER Mapping: Case#4 (Partial and Overlapping)



We need to create 3 tables:

كرواها

Person = (Person id, name, street, city)

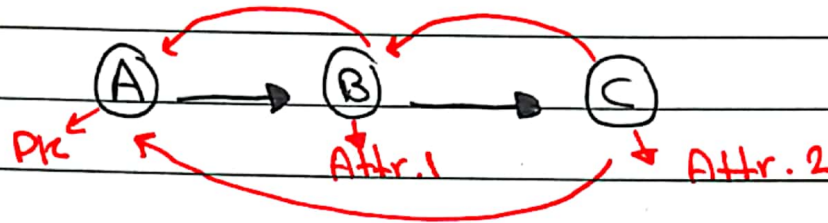
employee = (Person id, salary)

Customer = (Person id, credit_rating)

الناتج من الخوارزمية هو input لتطبيق الخوارزمية :-

Step 3:-

هذه هي الخطوة لـ Attribute في الجدول



إذا كان C يعتمد على B و B يعتمد على A فإن A يعتمد على C

النتيجة هي تبعية C على B إذا كان B


1. جدول Schema تبني على (الـ PK)

2. أي Schema لا يمكن أن تبني على attribute

3. أي Schema لا يمكن أن تبني على أكثر من Attribute

بشكل خاص Attribute (الـ PK)

في Schema يجب أن يكون لكل Attribute (الـ PK) و يجب أن يكون Attribute

الذي يعتمد عليه.  يعني تكلفنا فهمنا

Shyam Al-mawajdeh

Student

St_ID	St_Last	St_Address	St_City	St_POB	Course_No	Course_Name	Credits	Grade	F_ID	F_Last
0091111	Koko	XYZ Str.	Amman	1111	1902321	Database Management	3	C	2	Al-Zghool
					1903415	Database Tools	3	D		
0082222	Soso	ABC Str.	Amman	1212	1902321	Database Management	3	A	1	Al-Akhras
0076666	Lolo	Main Str.	Amman	3333	1903410	Information Systems	3	B	3	Al-Sayyed
					1901666	Computer Application	3	D		
0071234	Shosho	University Str.	Amman	9944	1903415	Database Tools	3	B	1	Al-Akhras
0088888	Dodo	WWW Str.	Amman	7878	1901666	Computer Application	3	C	2	Al-Zghool
2059922	Fofa	High Str.	Amman	3573	1902321	Database Management	3	A	3	Al-Sayyed

Exercise Solution

Baseline Table contains: *step*

**St_ID, St_Last, St_Address, St_City, St_POB,
Course_No, Course_Name, Credits, Grade,
F_ID, F_Last**

First Normal Form

In First Normal Form:

- (1) Primary keys are set
- (2) No repeating groups

Produced Tables:

<u>St_ID</u> , St_Last, St_Address, St_City, St_POB

<u>St_ID</u> , <u>Course_No</u> , Course_Name, Credits, Grade, F_ID, F_Last
--

Second Normal Form

In Second Normal Form:

- (*) No partial dependency ...

Produced Tables:

<u>St_ID</u> , St_Last, St_Address, St_City, St_POB

<u>Course_No</u> , Course_Name, Credits

<u>St_ID</u> , <u>Course_No</u> , Grade, F_ID, F_Last

Third Normal Form

In Third Normal Form:

(*) No transitive dependency ...

Produced Tables:

St_ID, St_Last, St_Address, St_City, St_POB

Course_No, Course_Name, Credits

StCourses F_ID, F_Last

StCourses St_ID, Course_No, Grade, F_ID

DDL Statements

In this lesson, we will learn how to write DDL statements to:

- create new table
- modify table structure
- drop a table

The general DDL structure for creating a table

- An SQL relation is defined using the **create table** command:

```
Constant ← create table [ → table name  
(  
  attribute name ← A1 D1, → data type  
  A2 D2,  
  ...,  
  An Dn,  
  integrity-constraint1,  
  ...,  
  integrity-constraint → ;  
);
```

- r is the name of the table (e.g. Student)
- each A_i is an attribute name (e.g. Student_ID) in the schema of relation r
- D_i is the data type of values in the domain of attribute A_i (e.g. varchar(40))

Create table syntax

```
Create table table_name
(
    column_name1    datatype(size),
    column_name2    datatype(size),
    column_name3    datatype(size),
    .....
    Integrity_constraints1,
    Integrity_constraints2
);
```

Create table syntax (con't.)

Integrity constraints in create table:

- not null
- **primary key** (*column_name₁, ..., column_name_n*)
- **foreign key** (*column_name₁*) **references** *table_name* (*column_name_n*)
- **primary key** declaration on an attribute automatically ensures **not null**

Create table Example

Create table *instructor*

```
(  
    ID          char(5),  
    name       varchar(20) not null,  
    dept_name  varchar(20),  
    salary     numeric(8,2),  
    primary key (ID),  
    foreign key (dept_name) references department (dept_name)  
);
```

Attributes of primary key and foreign key must be defined first in the attribute list.

For instance, in the above example we first need to define the attributes *ID char(5)*, *dept_name varchar(20)* first before we define PK and FK.

Updates to table structure

• Drop Table

• drop table *r*

//this will delete the table and its data

• Alter

• alter table *r* add *A D*
table

//this command to add a new attribute to the

- where *A* is the name of the attribute
- *r* is the relation name
- and *D* is the domain of *A* (data type)

• alter table *r* drop *A*
attribute

//this command to remove an existing

- where *A* is the name of an attribute
- *r* is the relation name

Drop table Ex.

To drop a table:

Drop table *instructor*;

To drop a table and its constraints: (when the table has a relationship with other tables via foreign key constraints)

Drop table *department* **cascade constraints**;

Try dropping table department without **cascade constraints**!

لر بدي احصا د كجود تابل

Drop table tab name;
↓
Table name

Alter table syntax

To add a column

Alter table *table_name* **add** ^{Column} *column_name* *datatype*;

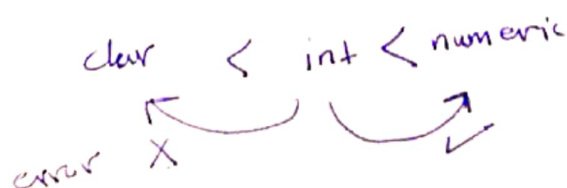
To drop a column:

Alter table *table_name* **drop column** *column_name*;

To modify a column:

Alter table *table_name* **modify** *column_name* *datatype*;

د ت
انغيره



نصير الحديل من اعطاك الدير
يسن تا نصير العكس