

## Mapping Rules (Relations)

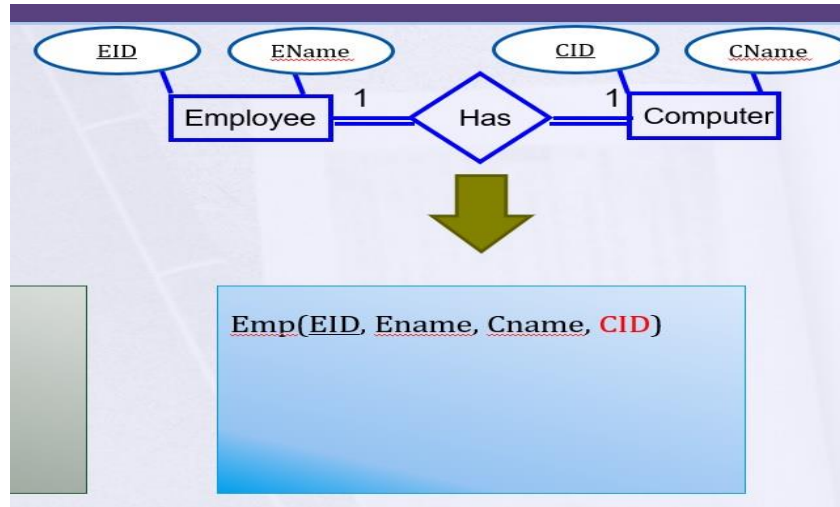
### \* Class 1 ( Cardinality 1 : 1 )

#### Case 1 : ( Total – Total ) :

- الـ two tables بيتجمعوا فى One Table

- الـ Pk الـ Whole Table بيكون أحد الـ PKs

( واحد منهم بس ) ( لأن already كل واحد فيهم Unique ومش هيتكرر فى نفس الجدول تانى )



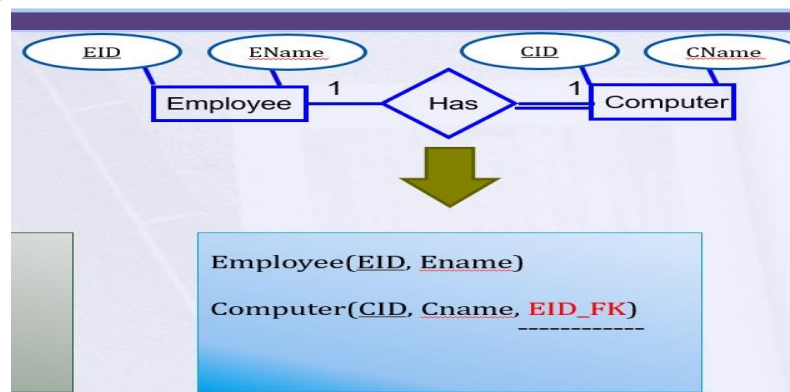
#### Case 2 : ( Total - Partial ) :

- الـ two tables بيفضلوا Two Separate Tables زى ما هما

- باخد الـ Partial Pk ويكون Fk فى جدول الـ Total

( منطقى إنه يتكرر أو يكون بـ Null لأن الـ employee may has a Computer )

بالتالى مش كل الـ Employees عندهم Computer لكن الـ Computer لازم يـ Belong الـ Employee



### Case 1 : ( Partial – Partial ) :

- Add new table (Third Table) الى ما هما وب

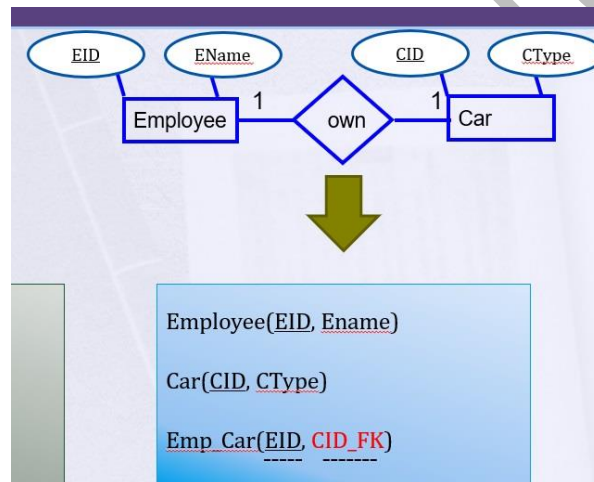
- ال Third Table بيحتوى على ال Two Pks

- كل PK يبقى Fk فى ال Third Table عشان يشاور على ال PK اللى جاى منه ( عشان يحافظ على ال Relation )

- واحد منهم يبقى ال PK لـ Third Table ( أى واحد لأن الإثنين هيضمنوا ال uniqueness لأنهم مش هيتكرروا لأن ال Cardinality عبارة عن One to One )

Employee may have one car (or not)

Each car may owned by only one Employee (or not)



## \* Class 2 ( Cardinality 1 : M )

- فى الـ 1:M أنا مهتم بالـ M Entity

### Case 1 ( Many is Total ) :

- يفضلوا Two Separate Tables زى ما هما

- بنقل الـ Partial PK ببقى FK فى الـ Total Table

( لأن الـ Department الواحد شغال فيه many Employees فعشان التكرار يبقى مسموح هخليه FK "معاه الـ values بتاعت الـ PK ولكن مسموح التكرار" فى الجدول اللى فيه Many )

مثال توضيح لـ data بعد ما تدخل :

### Employee table

Eid , Ename , DID

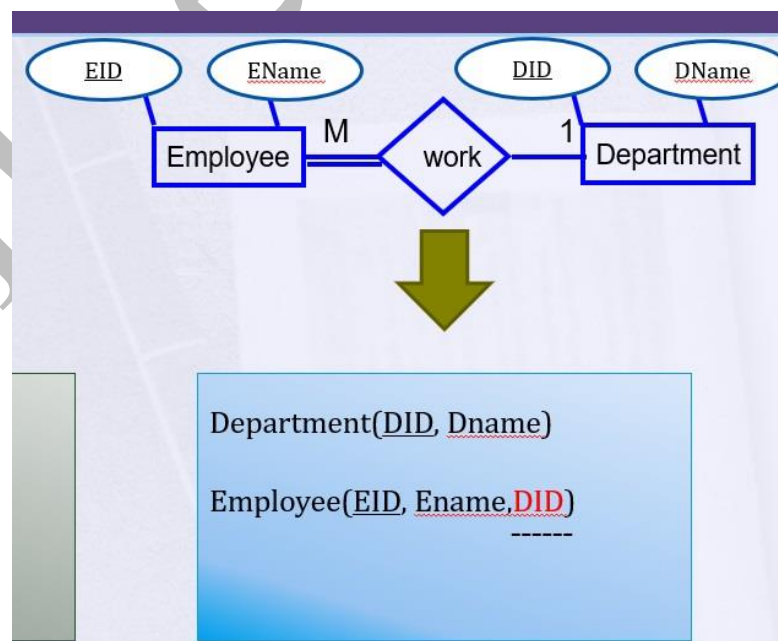
1 , Ahmed , 10

2 , Mahmoud , 10

- هيا دى الفكرة إن الـ Department رقم 10 مبيتكرش فى جدول الـ department

ولكن ممكن يتكرر فى جدول الـ Employee

ودا عشان إن الـ **Many** Employees work in **one** Department



## Case 2 ( Many is Partial ) :

- الـ Two Tables زى ما هما وبضيف Third Table بيحتوى على العلاقة  
- بنقل الـ Two PKs فى الـ third Table ويبقى كل واحد منهم Fk بيشاور على الـ PK اللى جاى منه

- الـ PK لـ Third Table بيكون الـ Pk الخاص بالـ Many Entity  
( ودا عشان يضمن الـ uniqueness لأن من الـ cardinality هنفهم إن

### Employee Works on one Project

One project contain ( is worked by ) Many Employees

فى الحالة دى الـ project هيتكرر لأنه فيه Many Employees لكن الـ employee الواحد بيشتغل فى project واحد بس  
مثال لـ data

### Proj Emp

EID , PID

1 , 10

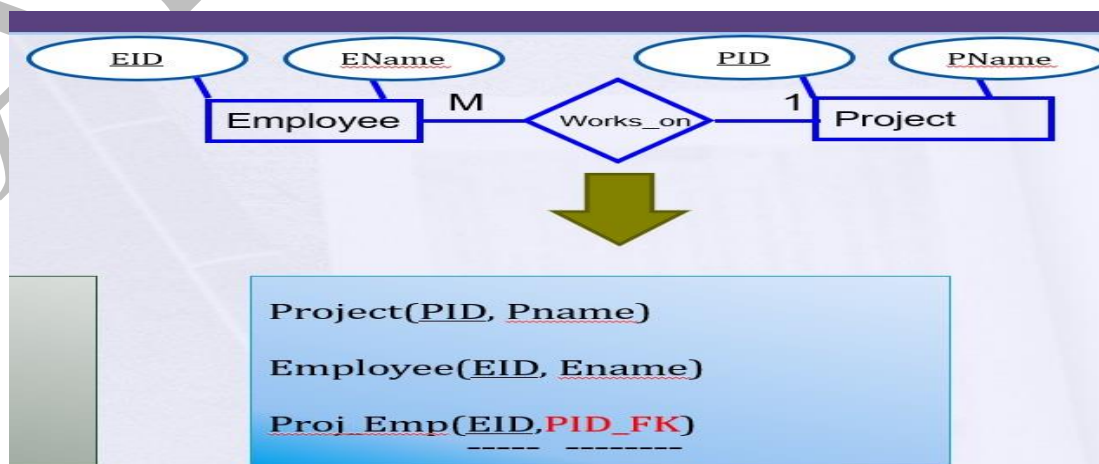
2 , 10

- عادى إن الـ project يتكرر عشان من الـ cardinality فى الـ case دى قال إن الـ Project بيتحتوى على Many Employees بيشتغلوا فيها

- لكن مينفعش أقول إن employee رقم 1 شغال فى Project 10 و Project 20

( لأن الـ cardinality عملت constraint على إنه يشتغل فى (One Project only

( يعنى مينفعش فى الـ third table اضيف row جديد فيه 30 , 1 مثلا - هيعترض ومش هيقبله فى الـ ( DB



### Class 3 ( cardinality M:M)

- فى حالة الـ Many : Many مش بهتم بالـ Participation

- بيفضل الـ two separate tables وبضيف عليهم Third table

- الـ third Table بيـ contain الـ Two Pks وكل واحد منهم بيكون Fk بيشاور على الـ PK اللى جاى منه

- الـ PK لـ Third Table بيكون Composite Key من الإثنين برضو

( لأن الإثنين many فطبيعى يتكرروا )

الـ Student may\ must takes Many Courses

والـ one Course may \ must Contains (is taken by) Many Students

فبالتالى عشان أضمن الـ uniqueness فى الـ Third Table وإن ميعترضش على الـ Data

هخلى الـ Two PKs يبقوا composite Key ودا أكيد مش هيتكرر

- مثال لـ data :

Stud\_course

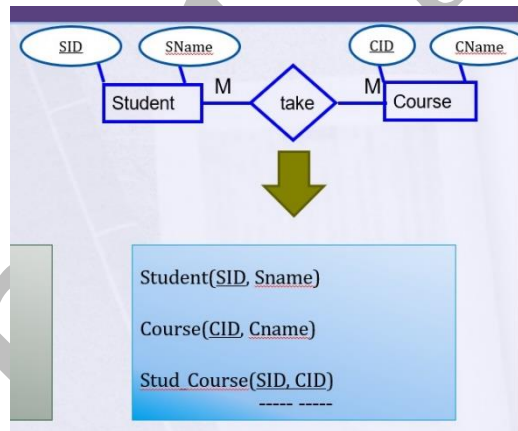
SID , CID

1 , a

1 , b

2 , a

2 , c



- هنا عشان الأثنين PKs كـ composite Key فمسموح إن أكرر الـ SID و الـ CID

لكن أكيد مش هكررهم الإثنين إن نفس الطالب ياخد نفس الـ CID ( مش منطقى )

لو كان مثلا بياخذها corrective وانتقال كذا فى الـ business كان هيبقى فيه تاريخ مثلا بيبقى معاهم فى الـ Composite Key يضمن إن الشخص لو خد نفس الكورس بيبقى أكيد خده وقت تانى ( ودا المنطقى )

## Class 4 ( ternary)

- فى حالة الـ Ternary

- بيبقى عندى الـ Three separate Tables

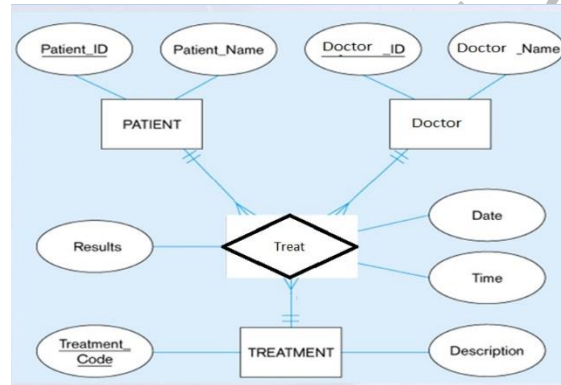
- بضيف عليهم 4<sup>th</sup> Table

- بيـ contain :

(1) الـ PKs من الـ three Tables بيكونوا Fks كل واحد منهم بيشار على الـ PK اللى جاى منه

(2) بشوف PK مناسب يضمن الـ uniqueness لو كان Composite بينهم Okey

لو غير كذا بختار أنا على حسب اللى يضمنلى الـ uniqueness



فى المثال دا

الحالة دى ممكن الـ patient ياخذ الـ Treatment من الـ Doctor كذا مرة عادى

فേഷان أضمن الـ uniqueness فيه date – time مستحيل يتكرروا

هاخذ أى Combination بحيث تضمن الـ uniqueness

مثلا الـ Doctor مع الـ date والـ time ( لأنه مش هيتكرر تانى لأنه مشغول فى الوقت دا بالتحديد مع حالة واحدة بس بيديها الـ treatment )

كذلك ممكن الـ patient مع الـ date time

بس مينفعش الـ treatment لأن ممكن نفس الـ treatment فى نفس الوقت يتكرر مع كذا حالة مريض من دكتور مختلف لمريض مختلف

## Class 4 ( unary)

ال unary إتعامل معاهم على إنهم logical two of same tables

وطبقت الحالات اللى فاتت فى ال Binary relations

Ahmed N. Esm