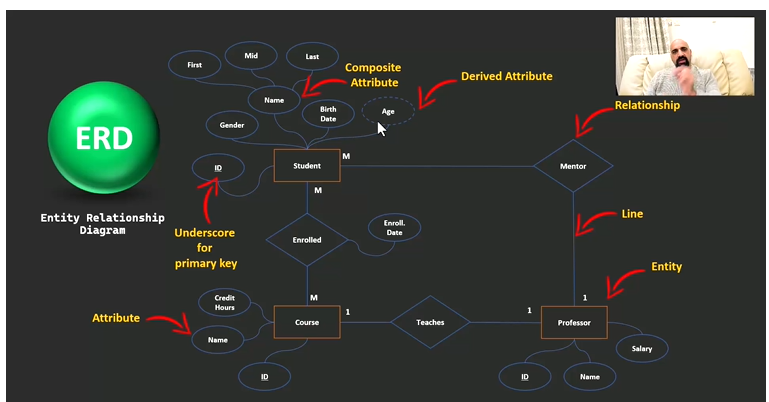
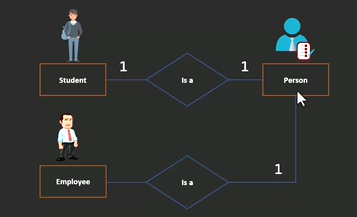
ERD Symbols



* Derived Attribute : هي عبارة عن خاصية يتم تمثيلها بنقاط ومارح أقوم بتخزينها بالداتابيز حيث يمكن حسبتها عن طريق خصائص أخرى مثل ال Age يمكن حسبته عن طريق ال (sysdate – birthdata)
* كل Entity (أي كل سطر بأي جدول) لازم يكون اله Primary Key
* Composite Attribute مثل ال Name لايتم تمثيلها بحد ذاتها ضمن الداتا بيز وانما يتم تمثيل أولادها

Relationships

We can have more than one relationship between two entities.



* السؤال السحري حتى حدد نوع العلاقة هو (ماهو ال Max) رايح جاي بين الجدولين

في كل العلاقات نحنا منحدد ال Maximum

كالتالي:

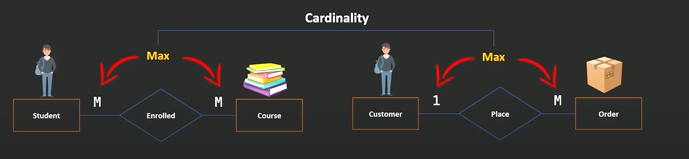
ماهو ال **Max courses** اللي **الطالب الواحد** اللي بيحسن يسجل فيه؟

ماهو ال **Max Students** اللي بيحسنو يسجلو **بكورس واحد**؟

* تحدديدنا لل Maximum بين العلاقات اسمه **Cardinality**

Cardinality vs Ordinality

1. Cardinality: تحديدنل لل Max

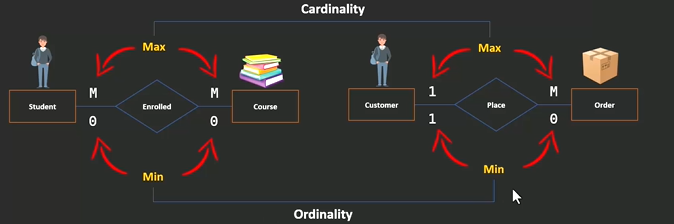


1. : Ordinality

تحديدنا لل Min

فيني يكون عندي طالب بدون كورس أو كورس بدون طالب(0 0)

فيني يكون عندي customer بدون order **لكن مافي order بدون customer (يعني مافيني اخلق order بدون customer)** (1 0)



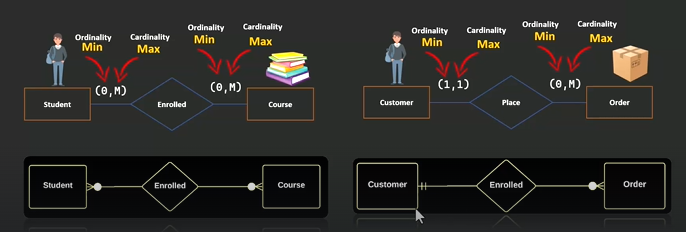
Cardinality بتفيدني في بناء العلاقات بين الجداول

بكتبهن بشكل مختصر كمايلي:

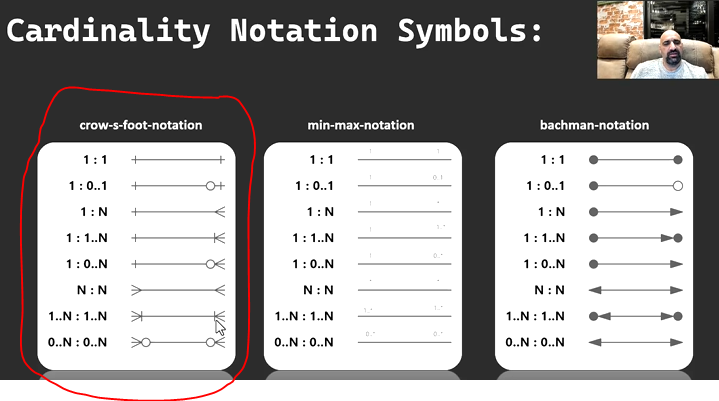


Crow's foot

ال Max (Cardinality)أقرب للEntity



جدا سهل التعامل مع النوع الاول (Crow’s foot)



Steps To Create ERD:

1. Entity Identification:

بدي استنتج الجداول (Entities) من ال Requirements

**أي شي بدي خزن عنه معلومة بالداتابيز هو Entity**

1. Relationship Identification:

هل يوجد علاقة بين الجداول **فقط بغض النظر عن نوعها**

1. Cardinality Identification:

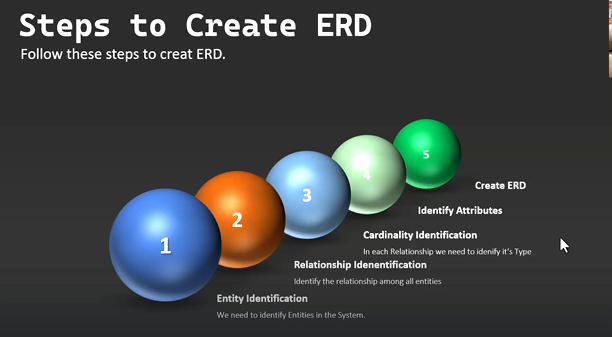
تحديد نوع العلاقة

1. Identify Attributes:

تحديد الاعمدة للجداول اللي أنشأناها بالخطوة الاولى

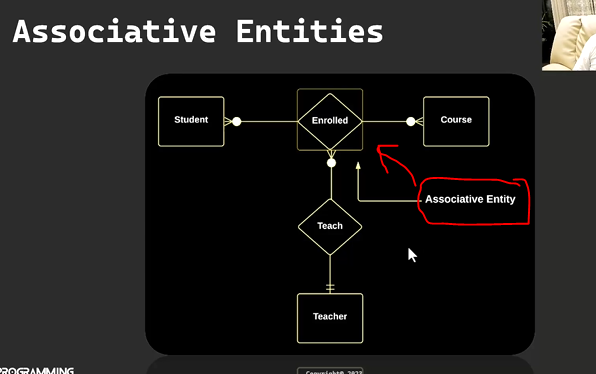
1. Create ERD:

بالعادة أنا برسم كل علاقة بملف لحاله -< بس جمعهن مع بعض بتنتج عندي الخطوة 5



Aggregation / Associative Entities

علاقة تبنى على علاقة

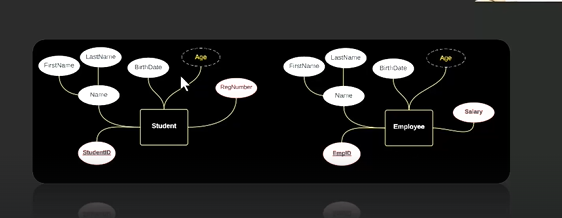


Generalization

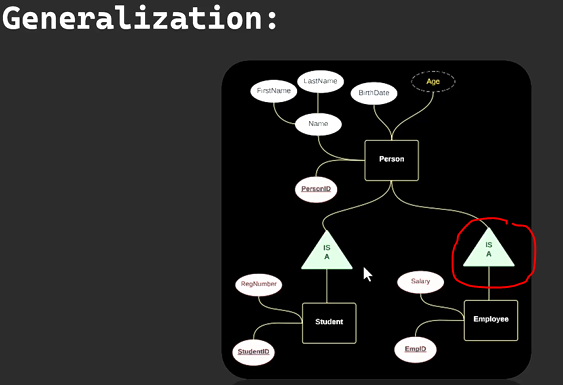
**بعمل List لكل شي خصائص مشتركة بين الكلاسن وبعملها Generalization**

**Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.**

في المثال التالي عندي معظم الخصائص مشتركة بين ال Student and Employee وهو يعتبر bad design



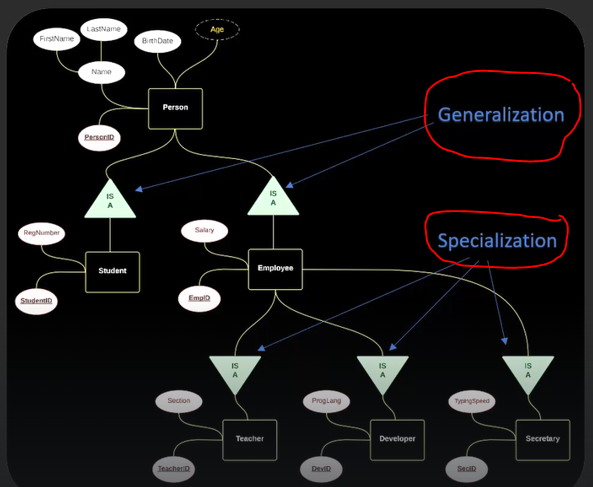
الحل يكون كالتالي:



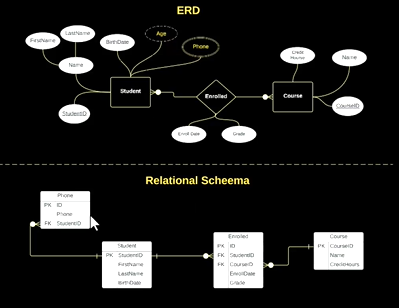
* **بمثل علاقة ال Generalization بالمثلث (رأسه فوق وقاعدته تحت) وبكتب ضمنه (IS A)**
* بقرأها من تحت لفوق (Student IS A Person) **Bottom-up approach**
* **بتحول من شي معروف Student لشي غير معروف Person**
* Person هو شيء General ومو **معروفة هويته لكن يتم تعريف هويته من ال Student او ال Employee**

Specialization

* بشتق منها Tables من نفس النوع (بخصصها)
* بخصصها يعني ال Employee ضله Employee لكن خصصته أكثر وصار (Teacher) يعني مابصير Employee خصصه لكورس!! بدو يكون من نفس النوع
* لكن ال Generalization جردته من كل شي وصار Person
* Top-down approach
* طبعا المثلث قاعدته للأعلى



ERD Diagram to Relational Schema



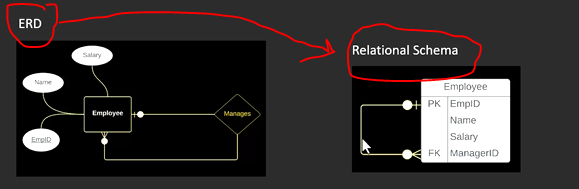
* أول شي كل Entity لازم تتحول لجدول عندي بالداتابيز

**وهذا الشي لايعني أنه هي فقط اللي رح أنشأها** لأنه ممكن أنشأ جداول اضافية أيضا ناتجة عن علاقات بين الجداول (**مثلا جدول كسر العلاقة بين ال Many\_To\_Many**) أو **ال Phone هي Multivalued Attribute بتنفصل بجدول لحالها**

* ال ERD هو Conceptual لكن ال Relational Schema (بنزل ليفيل بال Details)هي بتقلك بشكل واضح وصريح عن الجداول ولبعلاقات بينها اللي رح تنشأها فعليلا بالداتا بيز
* بنزل من ال ERD لل Relational Schema
* بال ERD + Relational Schema اسم ال Entity بكون مفرد لكن لما حوله لجدول بالداتا بيز بصير جمع لأنه يحتوي على مجموعة Entities (Entity Set)

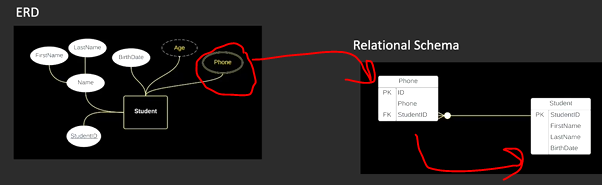
Convert ERD Relations to Schema Relations

1. Convert Self Join ==> Relational Schema



* **لاحظ انه لما حولت من ال ERD لل Relational Schema نتج عندي عمود ال Manager ID**

1. Convert Composite-Multivalued-Derived Attributes ==> Relational Schema



* ال (Name: Composite Attribute) :

أنا مابنقل الأب على الجدول وإنما أنقل الأبناء (بمشي على اخر جذر الها(First Name – Last Name لأنه ممكن تكون شجرة ومتشعبة كتير

* ال (Derived Class - Age) :

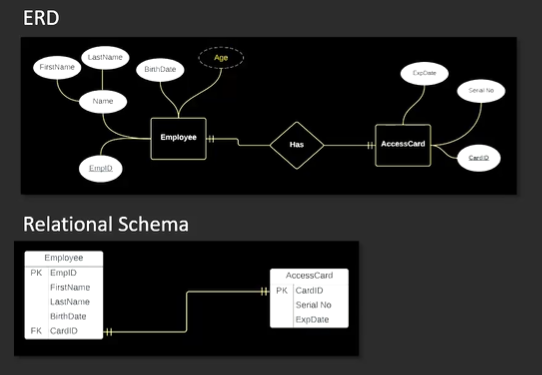
مابمثلها بالجدول لأني أنا هي بشتقها من عمود اخر B-Date عن طريق المعادلة (Sysdate – Birth date)

* ال (Multivalued Attribute - Phone) :

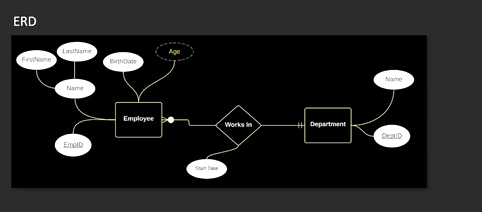
مابصير خلي الخلية تحتوي على أكثر من معلومة فبالتالي على السريع بروح:

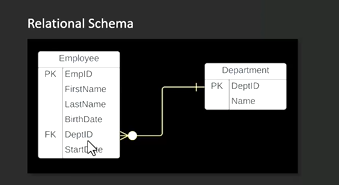
* **بعملها جدول وبسميه بإسمها**
* **بعمله ID(Pk) لانه أي Entity لازم يكون الها PK**
* **وبحط نفس ال Attribute Name- Name**
* **وبعمل عمود FK بنفس اسم الجدول الأب**

1. Convert One-to-One ==> Relational Schema:



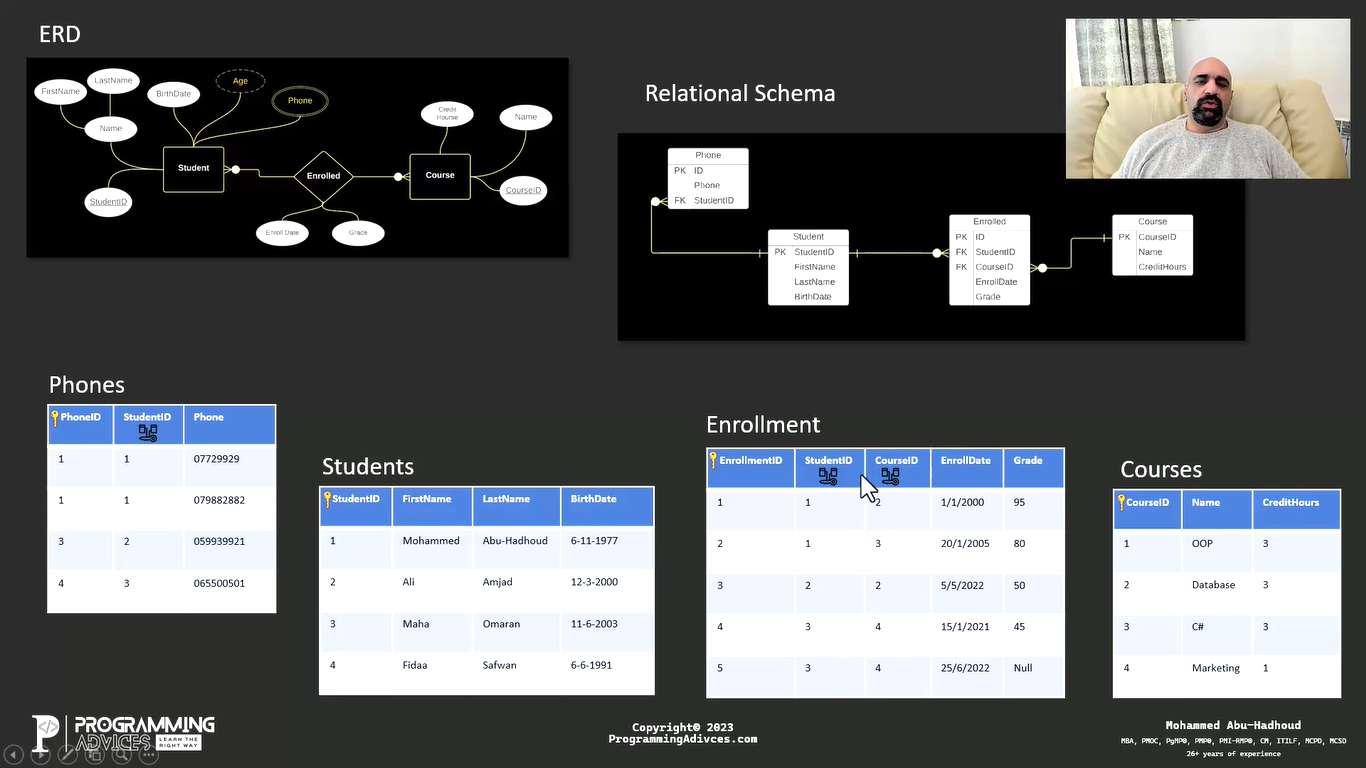
1. Convert One-to-Many/Many-to-One ==> Relational Schema:





* هنن نفس العلاقة (لكن حسب كيف بتقرأها)
* باخد ال PK من جهة جدول ال One (Departments)وبحطها ك FK في جهة جدول ال Many (Employees)
* أي علاقة تنتج عن الربط بين الجدولين (في مثالنا Start date ) **منحطه كعمود في جدول ال Many**

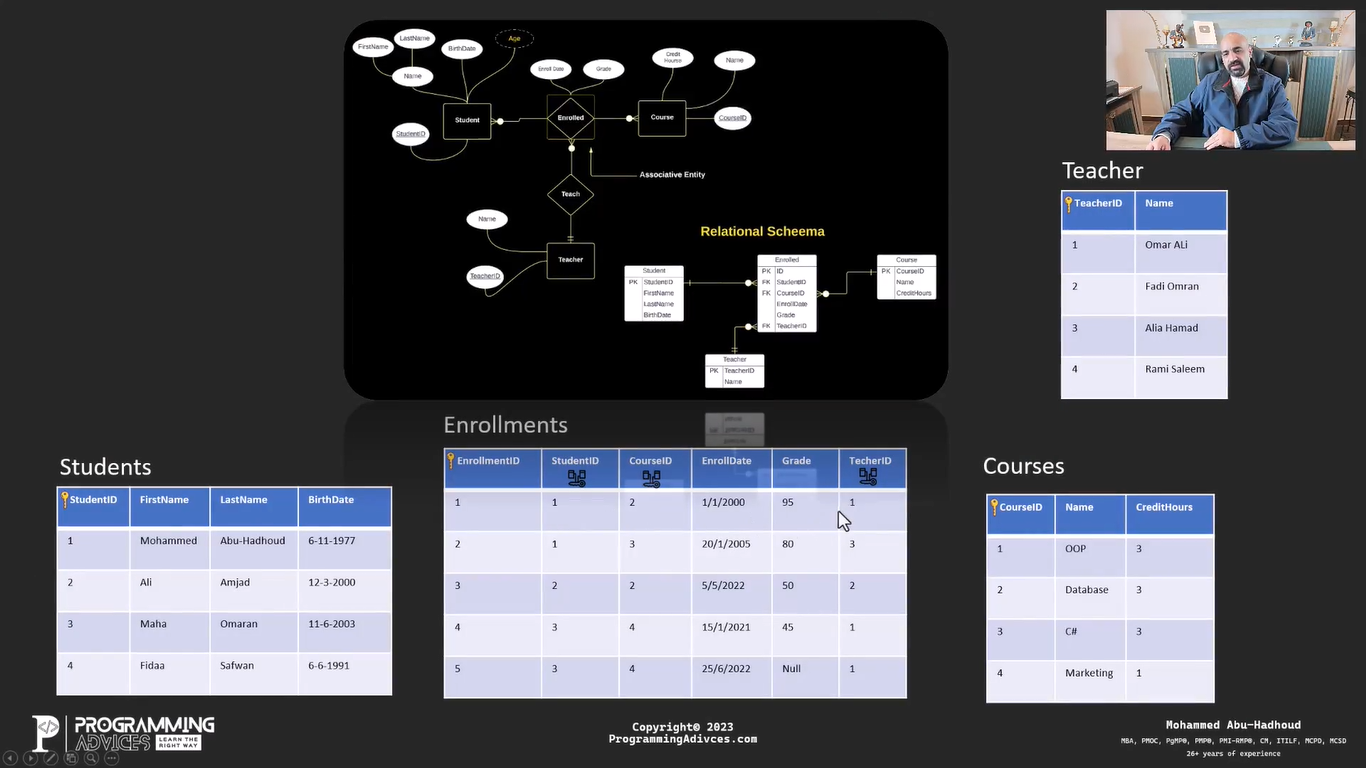
1. Convert Many-to-Many ==> Relational Schema:



* دائما في عندي جدول وسيط (كسر علاقة)
* بسميه ع اسم الخاصية (Enrolled)
* وظيفته يعمل Mapping (يعني بقلي الطالب اللي رقمه 10 بيدرس بالكورس رقم 5 بتاريخ 6/10/2023)
* بحطله Pk خاص فيه كالعادة (**لايفضل اني حط composite Pk Student\_id+Course\_Id**) هيك العالم بتشتغل بالعادة ولايفضل هذا الحل لانه اذا بتلاحظ مثال الطالبة **مها انها رسبت أول مرة في كورس ال Marketing ولو عندي Composite PK ماعاد فيني نزلها سطر جديد**
* بحط فيه ال Pks من الجدولين ك fks
* العلاقة مع الجدولين رح تكون One-To-Many
* بحط فيه ال Attributes

الناتجة عن هي العلاقة (Enroll Date, Grade)

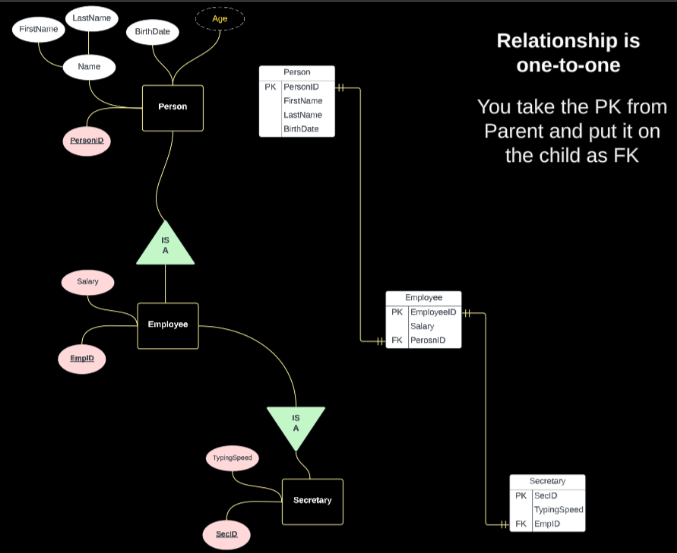
1. Convert Associative Entity to Relational Schema:



* **هي علاقة على علاقة**
* **هي دائما وفقط بين علاقة ال Many-To-Many على جدول كسر العلاقة**
* **بصير جدول كسر العلاقة عم يربطلي 3 جداول وليس جدولينس**
* كثير سهلة بتربطها ربط عادي
* هيك بتصير القراءة: **الطالب** محمد أبو هدهود عم **يدرس** كورس الداتا بيز عند **الأستاذ** عمر علي

Generalization and Specialization to Relational Schema

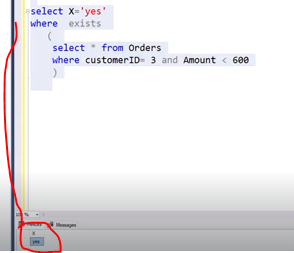
* Remember that the relationship is always one-to-One.
* You take the primary key from the parent entity and put it as foreign key in the child entity.



Views

* بدل مارجع الكويري كل مرة خلص بغلفها ب View
* Security أنا بعمل expose فقط لمعلومات معينة من الجدول بدل ما اعرض للمستخدم كل شي أعمدة

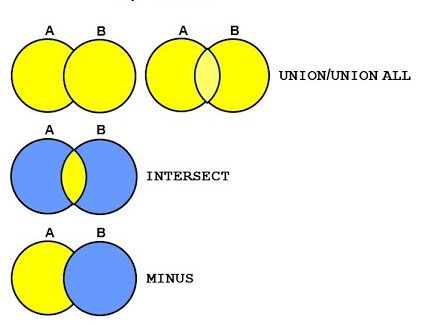
Exists



**مهما كانت ال Select الداخلية معقدة بمجرد أنها أعادة سطر واحد (يعني صارت True) رح يتم تنفيذ كامل جملة ال Select**

Set Operators

**Union بتجيب كل شي مع حذف التكرار Distinct**



Indexes

* الهدف منها تسريع استرجاع الداتا

1. Clustered Index:

* هو أسرع أنواع ال Indexes لأنه Physically موجود على نفس الجدول
* الجدول يحتوي على Cluster Index واحد
* بمجدرد ما أنشأنا Pk لحاله بينشألي Index نوعه Clustered Index
* كل مابضيف سطر جديد بروح على الجدول بدور على مكانه الصحيح وبحطه مشان هيك بالجدول ممنوع يكون عندي أكثر من Clustered Index واحد

1. Normal Index:

* أنا مابدي ابحث عن ال (ID) فقط مثلا أنا بستخدم Last Name

كتير بعمليات البحث

* لما أنشأ ال Normal Index بروح بأنشأ جدول يحتوي على الداتا مرتبة حسب ال Last Name فبالتالي لما بعمل Select على ال Last Name بروح بجيب الداتا من هاد الجدول اللي أنشأه
* فيني حط Normal Index قد مابدي
* ملاحظة:

أنا بعمل Indexes فقط للأعمدة اللي عليها Retrieval عالي بسبب أنه يحتاج الى مساحات أكبر لأنه عم يخزن الداتا مرتين بالاضافة انه يبطأ عملية ال Update + Add

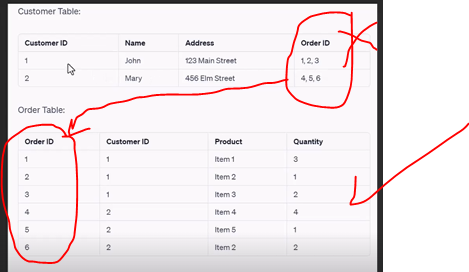
Normalization

بدي نظم الداتا Organization ضمن جداول وأقسمها بحيث مايكون عندي داتا مكررة وخلي ال Data Consistency فبالتالي بتصير أسهل باسترجاع و حفظ المعلومات

1. First Normal Form 1 NF:
2. لازم يكون عندك Pk
3. Atomic Values:

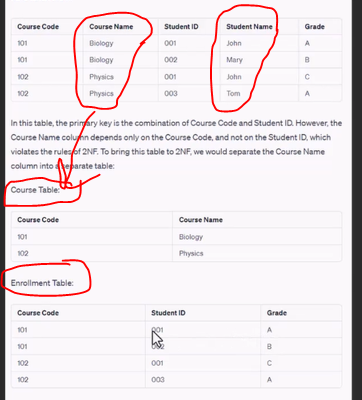
مالازم يكون عندي Multivalued Attributes (مثل مامر معنا بمثال ال Phonesيعني ال Column يحتوي على قيمة واحدة فقط فبالتالي نقوم بفصله بجدول اخر)

1. كل عمود لازم يكون اله اسم فريد (بشكل افتراضي ال Db Engine بيمنعك)



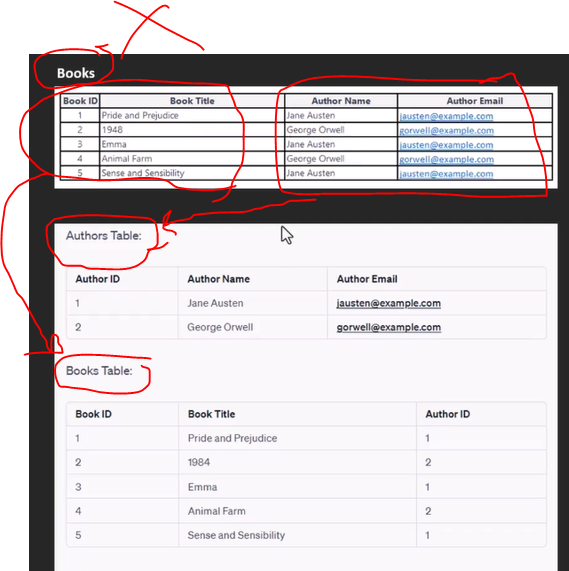
1. Second Normal Form 2NF:
2. اجباري لازم طبق ال 1NF حتى طبقه
3. No Partial Dependencies:

يعني أي Non Key Attribute لازم ترجع لمرجعية واحدة فقط (مثل ال Course Name وال Student Name أقوم بفصلهم بجدول لحالهن وبحط لكل واحد فيهن PK)



1. Third Normal Form 3NF:
2. لازم طبق ال 2NF وبالتالي ال 1NF
3. No Transitive Dependencies:

بتفهمها من الرسمة (بفصل ال Authors بجدول لحالهن )



**الخلاصة من موضوع ال Normalize هي سياسة فرق تسد (اني افصل معلومات ال Students عن معلومات ال Courses وهكذا....)**