Database Fundamentals Course

Day 1

* معظم الـ Applications وهنا بنتكلم عن الـ Enterprise Apps لازم يكون ليها DB بتحتوى على الـ Data الخاصة بـالـ System App

مثال لـ Apps ملهاش DB :

Paint – Microsoft Office

دى Apps مش محتاجة Database وتكلفتها عشان تخزن بياناتها

Database Lifecycle :

1- Analysis
2- DB Design
3- DB Mapping
4- DB Implementation
5- Application Design (GUI)
6- End User

1- Analysis :

- دى أول مرحلة فى مراحل DBMS Creation وفيها بيكون شخص مسمى وظيفته هو (System Analyst) ودا شخص عنده خبرة من ناحيتين (Business & Technichal) ودا بيفهم ال Requirements من الـ Clients وبيحولها لـ Requiremet Document الـ Developer يقدر يفهمها
- الـ Output بتاع المرحلة دى هو الـ Require Document
- الـ System Analyst بيحل مشكلة الـ Gap بين الـ Client و الـ Developer (حلقة الوصل بين الـ Bussiness Needs والـ Developer)

2- DB Design :

- مرحلة بيكون فيها الـ Input هو الـ Require Document اللى جاى من الـ System Analyst و بيجى دور الـ (DB Designer) ويحولها لـ ERD
- الـ ERD هو أختصار لـ Entity Relationship Diagram
- الـ Entity هو الـ System Component اللى محتاج أStore ليه Data

3- DB Mapping :

- عبارة عن Set of rules بتـ Apply على الـ ERD وبيكون الـ Output عبارة على الـ Actual Schema/ Database Schema

- لازم أرجع للـ System Analyst عشان أتأكد إن الـ Schema بتـ Meet الـ Requirements

- الشخص اللى بيقوم بالـ Role دى هو الـ (Database Designer)

4- DB Implementation :

- بيجى دور الـ DB Developer فى إنه يستخدم أحد الـ RDBMS Tools عشان يـ Create الـ Physical DB

- RDBMS (Relational Database Management System)

- Examples of RDBMS : SQL Server / Oracle / MySQL / Access …etc

- SQL (Structure Query Language)

- لما بتسطب أحد الـ RDBMS Tools بيتحول الجهاز لـ DB Server

5- Application Design :

- بيجى دور الـ Application Developer ودا شخص مسئول عن الـ Creation of GUI App عشان الـ End User يقدر يستخدم الـ DB System

- GUI (Graphical User Interface) : Web App / Desktop APP / Mobile APP

- الـ APP Developer هو الـ DB User لكن الـ Client هو الـ DB System User

- بيـ Deploy الـ GUI Application على Application Server

*** File Based System Vs DB System :**

- الـ File Based System : طريقة قبل ظهور الـ DB وهى عبارة عن إنك بتخزن الـ Data فى صورة Files بتستخدم فى الـ Small Apps وبتكون Low Cost

- أنواع الـ File Based System :

1- Delimited Files

2 – Fixed Width Files

- الـ Delimited Files بتستخدم Delimiterمعين زى (etc…. / - ,) بيفصل ما بين الـ Data

- الـ Fixed Width Files بيكون فيه عدد معين من الـ Bytes بتتحدد على أساسه الفواصل بين الـ Different Data

Problems/ Disadvantages of File Based Systems :

1- Difficult Search عشان هيلف على ال data one by one لحد ما يلاقى المطلوب حتى ولو كان فى أول الـ File

2- Data Duplication ودا بسبب ان مفيش حاجة تفهم الـ System إن الـ Data تم إدخالها قبل كدا ومفيش Control على المشكلة دى

3- Data Inconsistency ودا بسبب إن الـ Files are Seperated

4- No Relationships

5- No Data Integrity

6- No Data Quality

7- No Security

8- No Constraints

9- No Standard

10- Long Development Time

11- Low Performance

12- Manual Backup & Restore

13- No standard Format

- بس مميزاتها إنها مش محتاجة Cost وممكن تستخدمها فى الـ Small Apps

*** Advantages of DB Systems :**

- إختفاء كل مشاكل الـ File Based System

1- One standard

2- Data Quality

3- Data Integrity

4- Prevent Data Duplication Via PK

5- Relationships

6- Centralized & Shared

7- Security & Permission & Constraints

Automatic Backup & Restore -8

Consistency -9

- ممكن عيبها الـ Cost وبالتالى بنستخدمها فى الـ Enterprise Apps

---

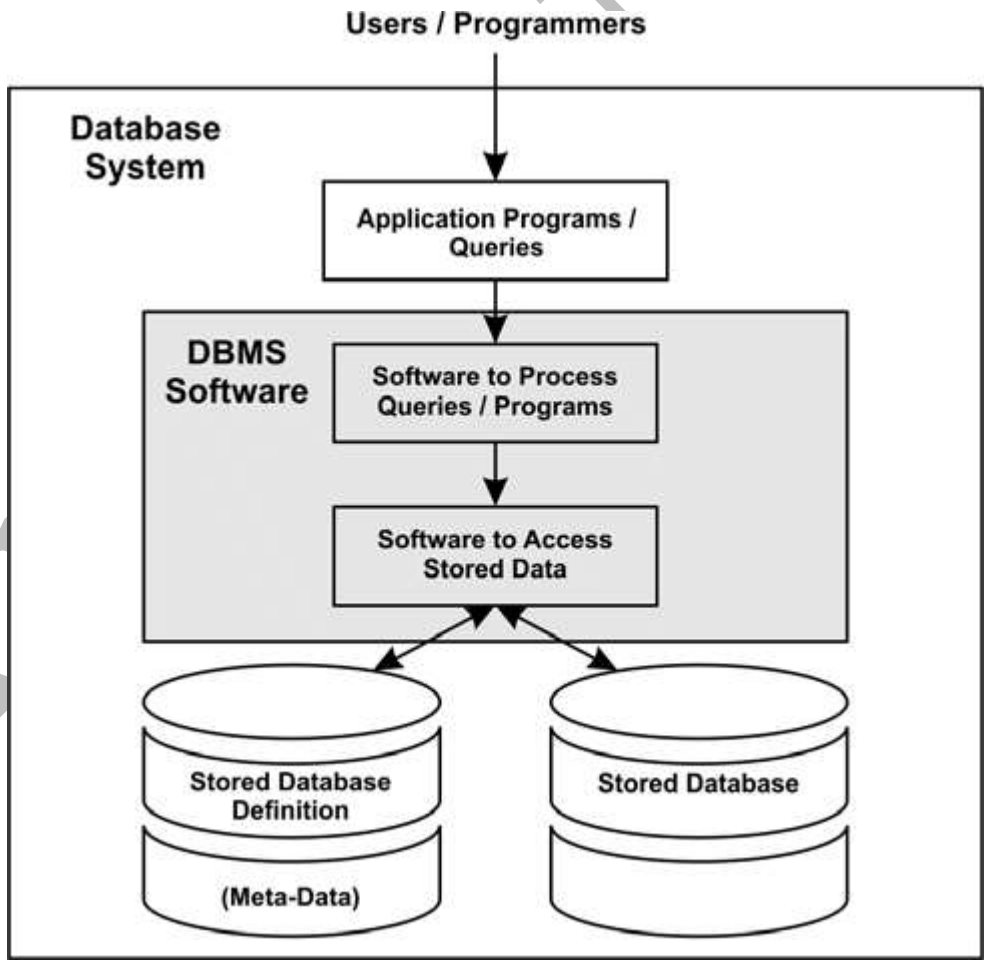
*** Basic Definitions :**

- Database : A collection of related Data

- DBMS : Software Tool that Create & Manage the Database

- DB System : (GUI App + DBMS)

*** Database Diagram :**



*** Database Users :**

1- DBA (Database Administrator) :

- دا شخص مسئول عن الـ maintenance و الـ Performance و الـ Security الخاص بالـ Database

2- System Analyst :

- دا شخص مسئول عن الـ Client Requirement Analysis وبيحولها لـ Requirement Document يقدر الـ Developer يفهمها من ناحية الـ Technical Wise

3- DB Designer :

- دا شخص مسئول عن الـ DB Design بيحول الـ Req Doc لـ ERD كبداية لـ DB Creation

- كمان مسئول عن الـ DB Mapping وهو تحويل الـ ERD لـ DB Schema

4- DB Developer :

- شخص عنده خبرة فى الـ SQL وبيستخدم الـ RDBMS Tools For DB Creation

- المسئول عن الـ Physical Schema Creation

5- Application Developer :

- دا شخص بيـ Design الـ GUI Applications عشان يعرض الـ DB System Features

6- BI & Big Data Specialist :

- شخص عنده خبرة فى مجال الـ ML و الـ Statistics بيقوم بعمل الـ Reports على الـ DB والـ Datawarehouse

7- End User :

- الـ Client اللى بيستخدم الـ DB System App فى النهاية

ERD Design

- طريقة لعمل الـ Modeling بيحول الـ Requirement Document لـ Graph يوضح الـ System Components و الـ Relationships بينهم

- بيبقى فيه PK بس

- عبارة عن :

1- Entity (System Components)

2- Attributes (Characteristics of Entity)

3- Relationships (Link of Entities)

- تمثيل الـ ERDعبارة عن Notations :

1- Rectangles (Entities)

2- Ellipses (Attributes)

3- Diamonds (Relationships)

- بيكون فيه Ellipses على الـ Diamonds ودى عبارة عن الـ Shared Attributes الموجودة فى Relationship معينة

- الـ Requirements Document بيبقى فيه :

- Names ودا بيمثل الـ Entities

- Verbs ودا بيمثل الـ Relationships

- ممكن يكون فيه كذا Relationships لـ One Entity بس بشرط يكونوا مختلفين مش نفس المعنى

1- Entity :

- أنواع الـ Entities :

1- Strong Entity

2- Weak Entity

- الـ Strong Entity هو اللى حذفه من الـ DB System مبيأثرش على Entity أخر

- الـ Weak Entity هو اللى حذفه جاى من حذف Strong Entity

- الـ Strong بيعتبر الـ Parent (Independent) أما الـ Weak بيعتبر الـ Child (Dependent)

- 99% من الـ Entities بتكون Strong

- مثال على الـ Strong Entity :

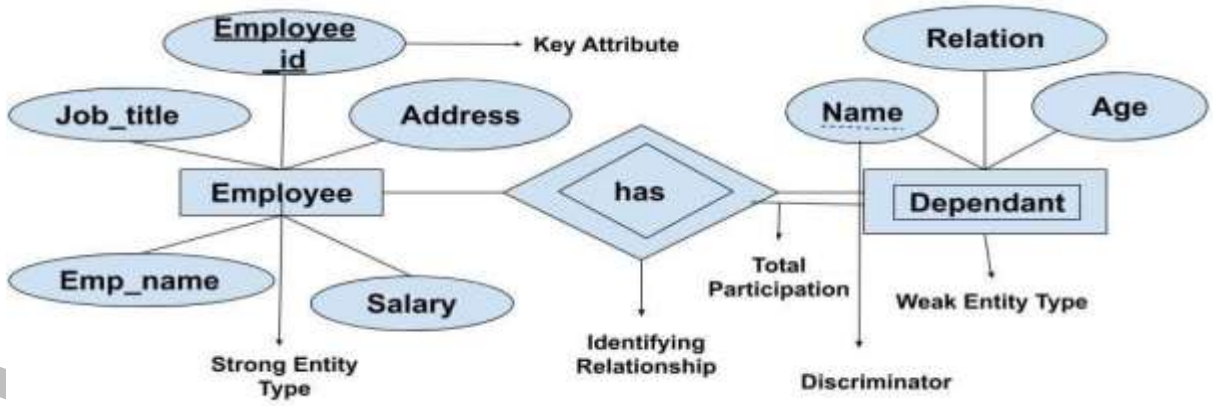
عندنا Two Entities اللى هما Employee و Department لو أنا حذفت أو رفدت الـ Employee دا مش معناه إنى حذفت الـ Department والعكس صحيح (من الأخر Independent)

- مثال على الـ Weak Entity:

عندنا Two Entities وهم Employee و Family لو أنا حذفت أو رفدت الـ Employee مفيش إحتاج لبيانات الـ Family الخاصة بيه (من الأخر بيـ Depend على الـ parent)

-تمثيل الـ Strong Entity بيكون عبارة عن Rectangle أما الـ Weak Entity بيبقى عبارة عن Dashed Rectangle

- الـ Strong ليه Primary Key أما الـ Weak بيبقى ليه Partial Key



Attributes

- أنواع الـ Attributes :

1- Simple Attribute

2- Composite Attribute

3- Deriven Attribute

4- Multivalued Attribute

5- Complex Attribute

*** Simple Attribute :**

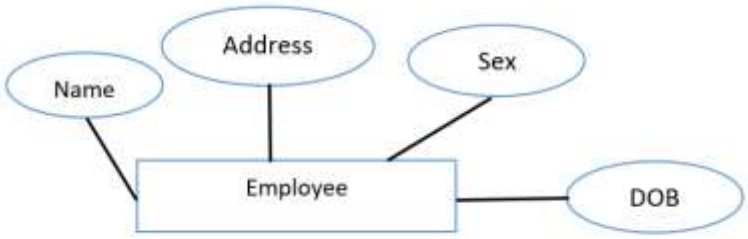
- ليه 3 شروط :

1- ميكونش بيتجزء

2- ميكونش بيتحسب فى الـ Run Time

3- ميكونش بيتكرر

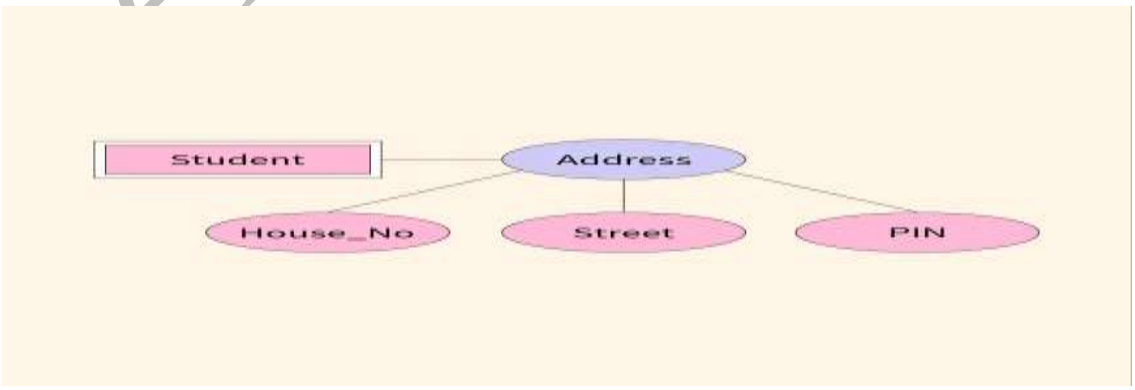
- تمثيل الـ Simple Attribute بيكون عبارة عن Simple Ellipse



*** Composite Attribute :**

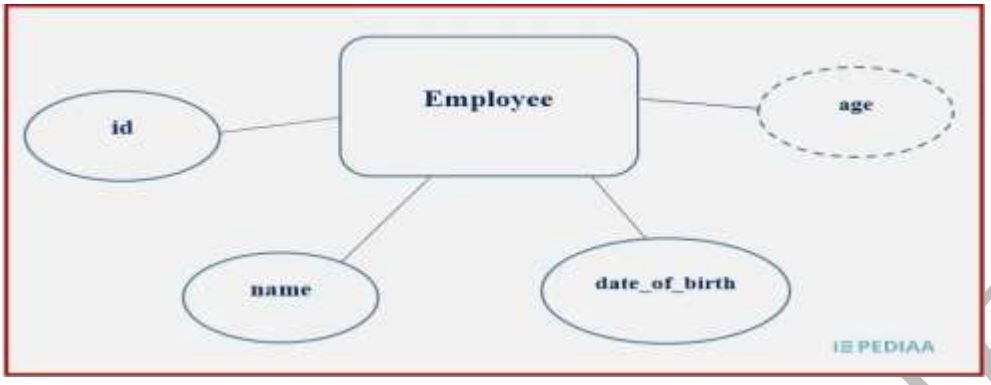
- بيكون أما يتجزأ <u>بشرط إن لما أجمع الأجزاء يدينى الحاجة الأصلية</u>

- تمثيله كالأتى :



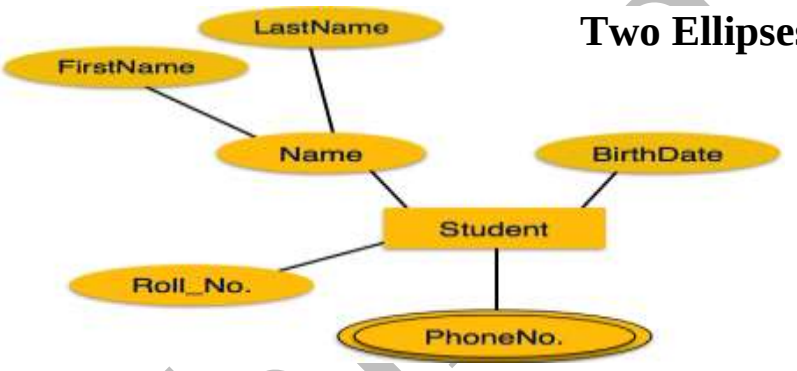
*** Derived Attribute :**

- بيكون الـ Attribute اللى بيتحسب فى الـ Run Time
- مثال : الـ Age بيكون عبارة عن طرح تاريخ اليوم من تاريخ الميلاد
- تمثيله بيكون Dashed Ellipse :



*** Multivalued Attribute :**

- بيكون الـ Attribute اللى ليه أكتر من Value
- مثال : ممكن الشخص يكون ليه أكتر من Telephone Number
- يمثل الـ Multivalued Attribute بـ Two Ellipses



*** Complex Attribute :**

- بيكون عبارة عن Multivalued & Composite
- بيكون أكتر من Value وكل Value بتتكون من Sub Values
- بيكون تمثيله كالأتى :

Relationship

- فيه 3 خصائص لازم أعرفها فى أى Relationship وهم :

1- Degree of Relationship

2- Cardinality of Relationship

3- Participation of Relationships

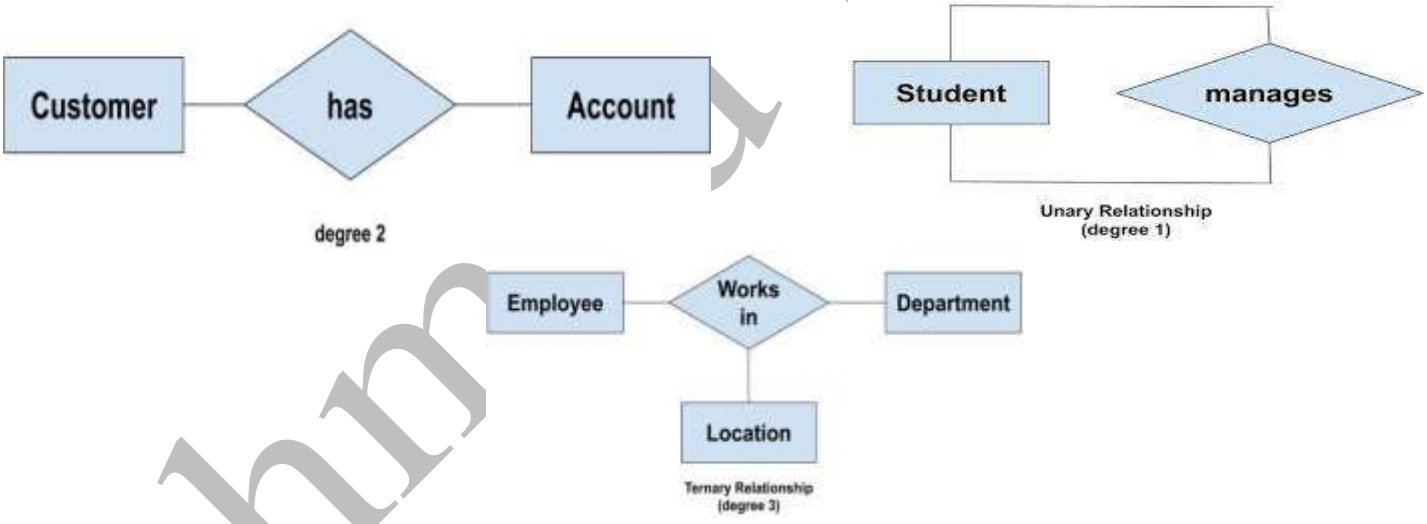
*** Degree of Relationship :**

أنواع الـ Degree of Relationship:

1- الـ Unary Relationship : ودى بتكون بين نفس الـ Entity بيسموها Self-Relationship

2- الـ Binary Relationship : ودى بتكون بين 2 Different Entities

3- الـ Ternary Relationship : ودى بتكون بين 3 or more Entities



*** Cardinality of Relationship :**

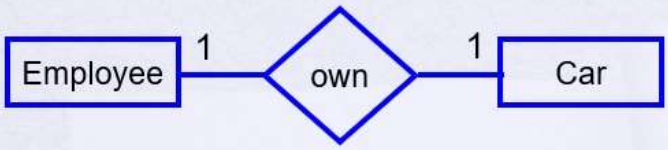
- بتكون عبارة عن نسبة الـ Entity فى الـ Relationship بالنسبة للـ other Entity وبتكون عبارة عن 3 حاجات :

1- One to One

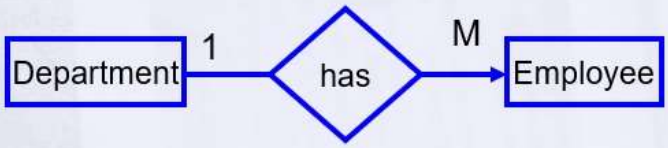
2- One to Many

3- Many to Many

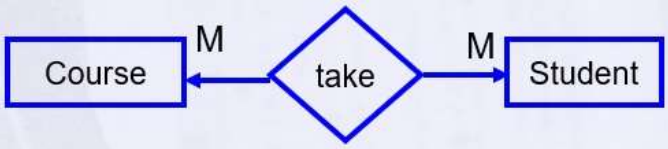
- One-to-One



- One-to-Many



- Many-to-Many



*** Participation of Relationship :**

بتكون عبارة عن إشتراك الـ Entity فى الـ Relationship وبتكون عبارة عن حاجتين :

1- Total Participation:

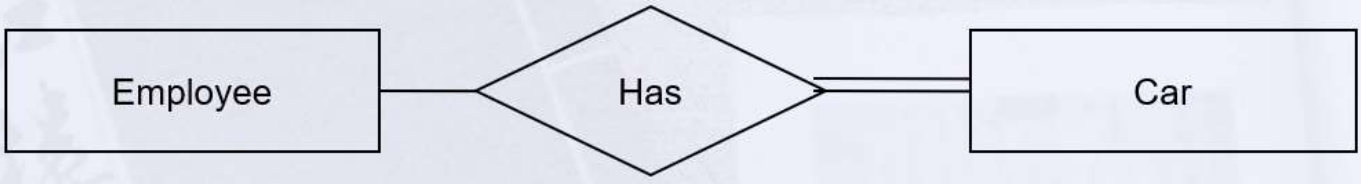
- الـ Keywords الخاصة بيها (one or more , must , mandatory , ….etc)

- تمثيلها بيكون Line أو Two lines

2- Partial Participation:

- الـ Keywords الخاصة بيها (zero or more , may , optional , ….etc)

- تمثيلها بيكون Dashed Line أو one line



-An Employee may have a car.
-A Car must be assigned to particular employee

أنــواع الـ Keys :

1-Primary Key

- هو الـ Attribute اللى استخدم عشان يعرف كل Instance من الـ Entity

- لازم يكون Unique & Not NULL

2- Partial Key

- الـ Primary key الخاص بالـ Weak Entity

3- Candidate Key

- الـ Attributes اللى تنفع تبقى Primary Key (محققة الشروط) ولكن مش مستخدمة as a PK

- الـ Primary key واحد من الـ Candidate Keys

4- Super Key

- هو الـ Superset of Candidate Keys

5- Alternate Key

- هو الـ Attribute\s اللى محققة الـ Primary Constraints

- الـ Alternate Key هو نفسه الـ Candidate (من غير الـ Primary Key)

6- Composite Key

- لما يكون الـ primary Key عبارة عن 2 or more Attributes

7- Foreign Key

- بيستخدم فى الـRelationship بين الـ Tables كربط بينهم

- بيبقى تمثيل لـ Primary Key من Table تانى

8 – Artificial Key

- بيكون Key من إختراعك

- بتلجأ ليه لما تلاقى الـ Primary Key بقا Large ومعقد