**DATABASE**

**Relational Database:**

Any database in table/entity (rows/record/relation and columns/field/attribute).

**Normalization:**

Reduce data redundancy, save space in hard disk, remove Nulls.

**ERD:**

* Entity Relationship Diagram
* Represent the relationship between entities
* Contains metadata Only(entities and attributes )
* Rectangle🡪 Entities
* Diamond🡪 Relationship
* Ellipse🡪 Attribute

**Strong Entity:**

* Entities don’t depend on others
* Must have primary key

**Weak Entity:**

* Entities depends on others
* Represents by Double Rectangle
* Course and labs are 2 entities🡪 if course deleted 🡪 there is no labs🡪 labs is a weak entity
* Have partial key🡪 primary key of weak entity

**Attributes:**

Characteristic of attributes:

* **Simple Attributes**: لا يتجزأ ولا يحدث له اي عمليات حسابيه🡪 (Name, City,…)
* **Composite Attributes**: يتجزأ (Name is firstname and lastname)
* **Derived Attribute**: يتجزأ و يتحسب فى الرن تايم Such (age, NetSalary,…), represented by dashed ellipse
* **Multi-Valued Attribute**: نفس الشخص ليه كذا رقم موبايل و كذا عنوان , represented by double ellipse
* **Complex Attribute**: Multi-Valued + Composite

**Relationships:**

Diamond represents the relationship between entities.

**Types/ Degree of Relationship:**

* **Unary/ Recursive** : relationship for same entity (Entity of the same type[manager is an employee])
* **Binary** : relationship for 2 entities
* **Ternary**: relationship for 3 entities

**Cardinality of Relationship:**

* **One- to-one:**
* **One- to-Many:**
* **Many-to-Many:**

**Participation Constraints:**

* **Total Participation:** Must in the statment
* **Partial Participation:** may, Optional in the statment

**Table 1**

+------------+------------+---------+

| StudentID  | GivenNames | Surname |

+------------+------------+---------+

| 12345121   | Sarah      | Doe     |

| 12345303   | Susan      | Smith   |

| 12345678   | John Paul  | Bloggs  |

| 12345876   | Susan      | Smith   |

+------------+------------+---------+

**Table 2**

+------------+-----------------------+------+-----+------+

| StudentID  | CourseName            | Year | Sem | Pctg |

+------------+-----------------------+------+-----+------+

| 12345678   | Data Science          | 2019 |   2 |   72 |

| 12345121   | Programming 1         | 2020 |   1 |   87 |

| 12345678   | Computing Mathematics | 2019 |   2 |   43 |

| 12345678   | Computing Mathematics | 2020 |   1 |   65 |

| 12345121   | Data Science          | 2020 |   1 |   65 |

| 12345876   | Computing Mathematics | 2019 |   1 |   75 |

| 12345876   | Programming 1         | 2019 |   2 |   55 |

| 12345303   | Computing Mathematics | 2020 |   1 |   80 |

+------------+-----------------------+------+-----+------+

**Key:**

Column or set of columns that return one unique value/row.

الكولوم اللي يحتوي علي قيم فريده بسسس

**Table 1: StudentID is a Key**

**Table 2: (StudentID, CourseName, Year and Sem) is a Key**

**Candidate Keys:**

Table that have more than one key/ columns that can be considered as keys.

مجموعات الكولومز اللي ينفع تبقي كيز

**Table 1: (StudentID) and (StudentID, GivenName) and (StudentID, SurName) and (StudentID, GivenName, SurName) all are Keys**

**Composite Key:**

كذا كولوم مع بعض ينفعوا برايمري كي

**Primary Key / Super Key:**

* Key that have minimum number of columns from candidate keys🡪 مجموعه الكيز اللي فيها اقل عدد من الكولومز
* Unique
* Can’t be Null
* Represented by underline
* Key / Column that connect tables with others
* **Table 1: (StudentID) is a Primary\_Key**

**Foreign Key:**

* Column shared between tables and it’s a primary key in another table

هو كولوم موجود في التيبل ده و في التيبل الاخر علي انه برايمري كي

* Not necessary to be same name of primary key, but must be same data type
* Accept Nulls
* Represented by dashed line

**RDBMS:**

* Relational Database Management System.
* It’s a software app designed to store, manipulate and manage data.

**OLTP:**

* Online Transaction Processing 🡪 بسجل كل حاجه في الداتا بيز
* CRUD Operation(Create, Read, Update, Delete)
* Important to be Normalized(Best Performance)
* Managed by RDBMS.

**OLAP:**

* Online Analytical Processing🡪 الغرض هنا اني اعمل ريبورت من الداتا اللي عندي
* Get data and make report(Read ONLY) 🡪 Expensive
* Important to be Denormalized(اخليها في جدول واحد لان الريد اوبريشن كده كده مكلفه فمش هيبقي زياده مساحه كمان)
* Managed by Data Warehouse(Optimized on Write-Once Read-minitanse)

**DDL:**

* Data Definition Language
* اي حاجه بتعدل في الاستراكشر بتاع الداتا نفسها
* create, modify, and delete database

**DML:**

* Data Manipulation Language
* التعامل مع الداتا نفسها اعدل فيها و احط و اعدل و امسح و اسيليكت
* Manipulate Data through CRUD operation

**DQL:**

* Data Query Language
* Select + anything(joins, having, ….)

**DCL:**

* Data Control Language
* Permisions
* السماح و عدم السماح

**Declarative:**

* اقولك انا عاوز ايه بس مقولكش تعمله ازاي

**Costraints:**

* **Not Null:** columns don’t contain any null values
* **Unique:** columns don’t contain duplicated values
* **Primary Key:** not null+ unique and one PK for table consists of 1 or more columns{primary key (columns)}
* **Foreign Key:** make links between tables , foreign key refrences(primary key)
* **Check:** Ensures that values in column satisfy the condition, {check(condition)}
* **Default:** add default value to column, added to all new records if these haven’t values, {default+value}
* **Create Index:** to retrieve data very quickly, {create index+اسمه +on +table name(columns)}

**SQL DB Statements:**

1. Create database+ اسمها
2. Drop database+ اسمها
3. Backup database+ اسمها + to disk = ‘path’
4. Create table + اسمه +(attribute , datatype, constraints);
5. Drop table + اسمه🡪 (drop table itself)
6. Truncate table + اسمه🡪 (drop data but table still exists)
7. Alter table+اسمه + one of the following:

* Add (attribute , datatype, constraint)
* Alter column+(attribute , datatype, constraints)
* Drop column (attribute )
* Rename column (old column to new column)

**Select:**

Such print, used to read data and print it such:

1. SELECT 'Hello, SQL'; -- single quotes for strings
2. SELECT 'My', 'name', 'is', 'Ahmed'; -- select multiple values
3. SELECT 1, 'Test'; -- select different data types

**Alias:**

اسم كستعار يعنى مثلا لما كنا بنطبع حاجه مثلا زى الاسم بايثون كنا بنعمل كده

Print(Name) --- فكنا بنعملها بشكل احسن زى كده --🡪 print(“Name is”, Name)

Alias كده بالظبط بس بستعمب معاها كلمه (AS)

SELECT 'Hello, SQL' AS greeting;