



# MAGIC CODE INSTITUTE



Microsoft®  
**SQL Server**



**MAGIC CODE INSTITUTE**

# Data Modelling



# Primary key & Foreign key

**Primary key** (PK - khóa chính): được dùng để đảm bảo 1 record trong 1 bảng là duy nhất. PK có thể gồm 1 (single PK) hoặc nhiều cột (composite). PK **không** nhận giá trị NULL.

**Foreign key** (FK - khóa ngoại): được dùng để trỏ tới (refer to / tham chiếu tới) PK của 1 bảng khác. Chính các khóa ngoại này giúp tạo nên mối liên hệ (relationships) giữa các bảng với nhau.

```
CREATE TABLE production.products (  
    product_id INT IDENTITY (1, 1) PRIMARY KEY,  
    product_name VARCHAR (255) NOT NULL,  
    brand_id INT NOT NULL,  
    category_id INT NOT NULL,  
    model_year SMALLINT NOT NULL,  
    list_price DECIMAL (10, 2) NOT NULL,  
    FOREIGN KEY (category_id) REFERENCES production.categories (category_id) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (brand_id) REFERENCES production.brands (brand_id) ON DELETE CASCADE ON  
    UPDATE CASCADE  
);
```



CASCADE: nếu có thay đổi ở parent table (bảng được ref tới) thì thực hiện thay đổi ở bảng này tuân.

# Relationships

[Sample blog post](#)

**Relationships:** mối liên hệ giữa các bảng với nhau, được thể hiện bằng tỉ lệ giữa PK của bảng parent và FK của bảng child.

Trong RDBMS (relational....) có 3 loại relationships:

- 1-to-1 (one-to-one) (incl. Self-joining relationships): thường là dimension tables với nhau
- 1-to-n (one-to-many): dimension - fact tables
- n-to-n (many-to-many): cooperative (authors - books, students - lectures...)

Ví dụ về relationships: [Sample ERD](#)



# Data modelling

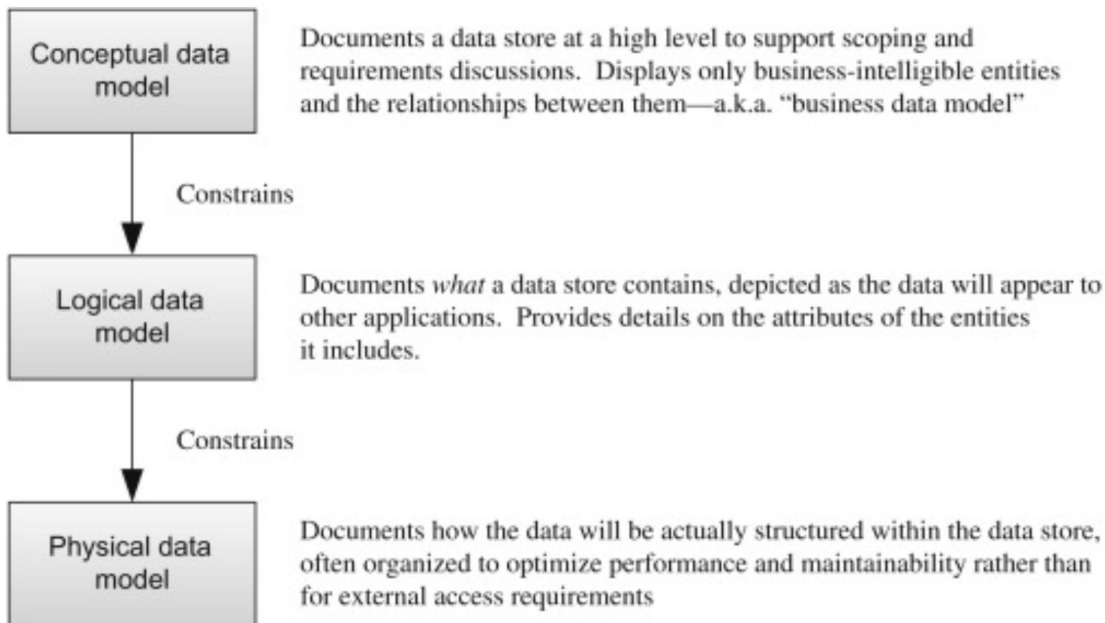
**Data model:** cấu trúc lưu trữ của các dữ liệu.

**Data modelling:** công việc tổ chức, sắp xếp việc lưu trữ của dữ liệu.

**Mục đích:** đáp ứng yêu cầu, dễ vận hành & bảo trì, tối ưu hóa nguồn lực về query & **chi phí**

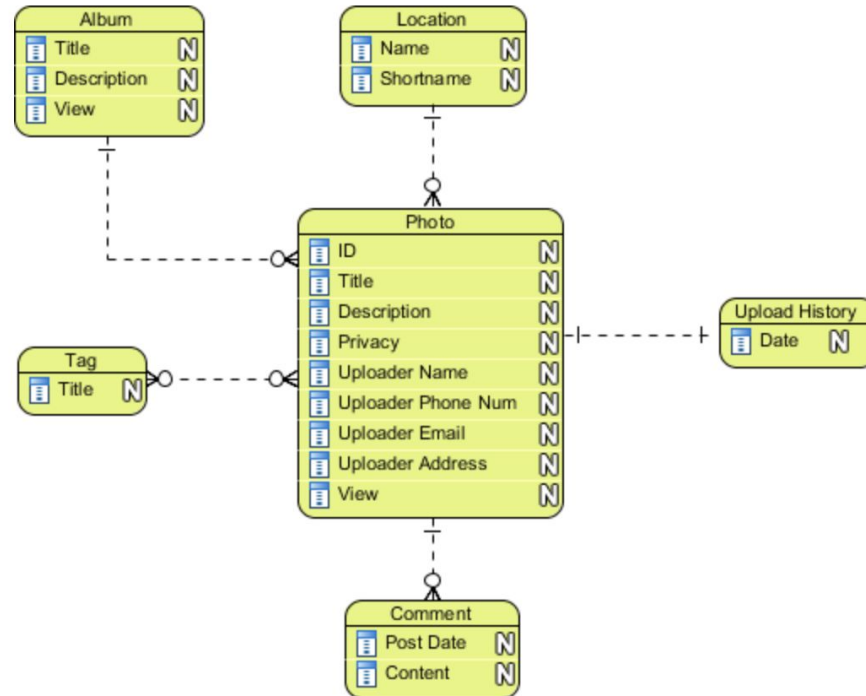
Một số hình thái của Data Model:

1. **Star-schema**
2. Hierarchical Model (sitemap, food recipes... )
3. Network Model (Student & Department & Classes)
4. Entity-Relationship Model
5. Relational Model
6. Object-Oriented Data Model
7. Object-Relational Data Model
8. Flat Data Model
9. Semi-Structured Data Model
10. Associative Data Model
11. Context Data Model



# Data modelling

Conceptual ERD models information gathered from business requirements. Entities and relationships modeled in such ERD are defined around the business's need. The need of satisfying the database design is not considered yet. Conceptual ERD is the simplest model among all.



Conceptual ERD example

# Data modelling

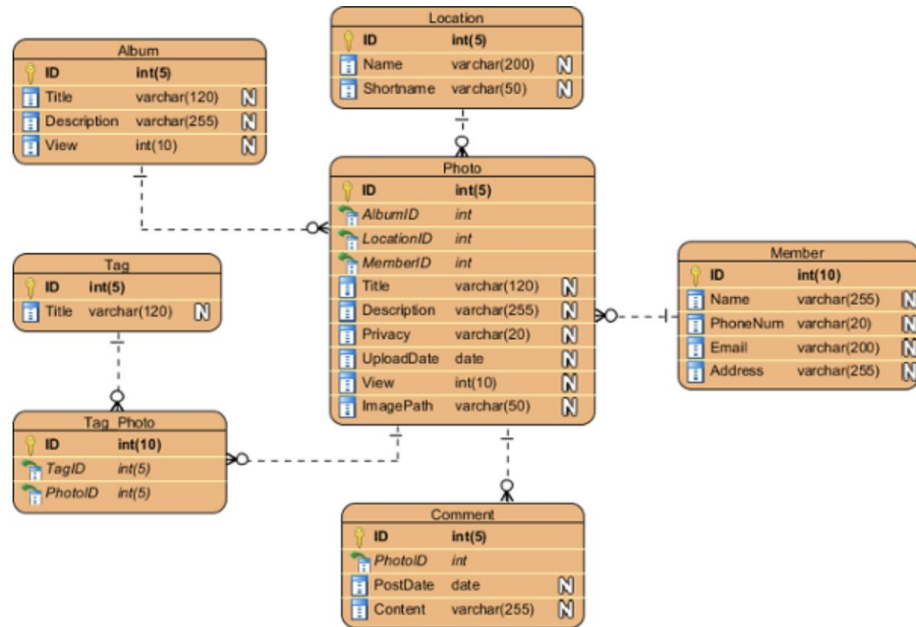
Logical ERD also models information gathered from business requirements. It is more complex than conceptual model in that **column types are set**. Note that the setting of column types is optional and if you do that, you should be doing that to aid business analysis. It has nothing to do with database creation yet.



Logical ERD example

# Data modelling

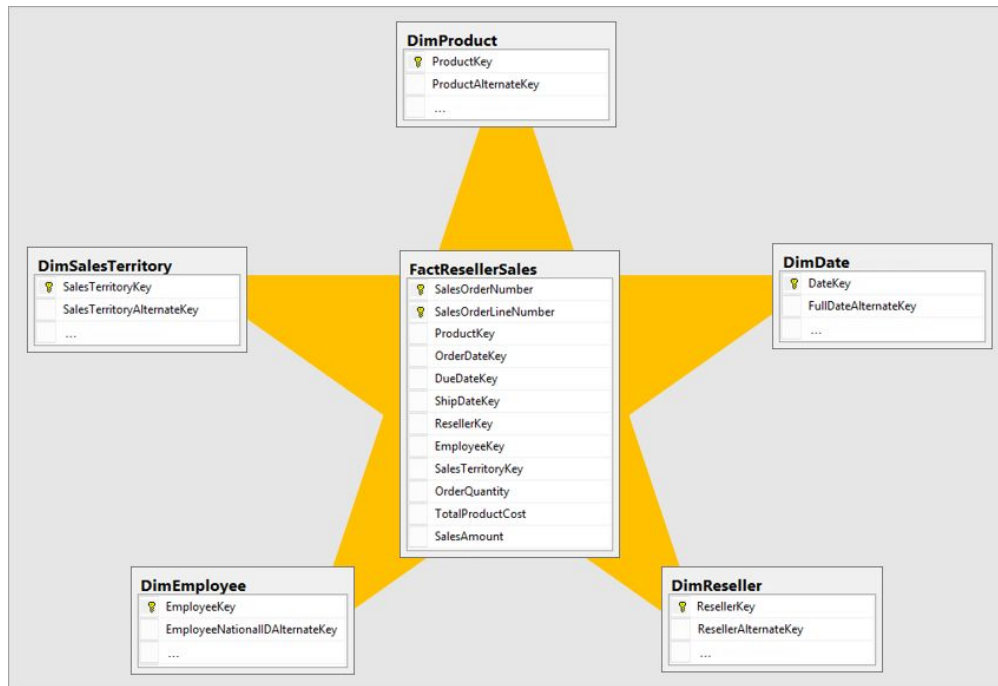
Physical ERD represents the actual design blueprint of a relational database. It represents how data should be structured and related in a specific DBMS so it is important to consider the **convention and restriction** of the DBMS you use when you are designing a physical ERD. This means that an **accurate use of data type** is needed for entity columns and the use of **reserved words has to be avoided** in naming entities and columns. Besides, database designers may also **add primary keys, foreign keys and constraints** to the design.



Physical ERD example



In computing, the star schema is the simplest style of data mart schema and is the approach most widely used to develop data warehouses and dimensional data marts. The star schema consists of **one or more fact tables** referencing **any number of dimension tables**. The star schema is an important special case of the snowflake schema, and is more effective for handling simpler queries.



Hãy chọn 1 lĩnh vực và thiết kế database để mô tả các nghiệp vụ của lĩnh vực đó.

Ví dụ:

- Banking → làm sao để quản lý các chi nhánh? Nhân viên của chi nhánh? Quản lý số lượng thẻ / tài khoản/ nợ được phát hành/ giải ngân?
- Marketing → làm sao để quản lý hiệu quả của nhiều team? nhiều campaign? Project? ad quảng cáo? Làm sao biết runrate hiện tại ntn?
- FMCG → làm sao để quản lý số lượng hàng hóa trên thị trường? làm sao biết khi nào cần liên hệ với nhà cung cấp để restock?
- ....

# BTVN

1. Hãy làm tiếp bài thực hành ạ =))
2. Hãy thử import dataset mẫu vào DB local ạ :D [\[link guide\]](#)
3. Đọc thêm về Indexes, cách tính chi phí của các database (Vd: Google BigQuery, AWS RDS, Redshift, Spectrum, Athena...)
4. (Dành cho các bạn DE/ DBA) đọc thêm về Query plan (EXPLAIN)





**MAGIC CODE INSTITUTE**

**THANKS FOR LISTENING!!!**