**- Tại sao Website không hoạt động hiệu quả:**

+ Xây dựng xong bị bỏ quên, không “marketing” website. Marketing qua mạng là thực hiện các hoạt động quảng bá 1thông điệp tới nhóm đối tượng quảng bá dựa trên các công cụ email, www…

+ Chất lượng website thấp: nội dung nghèo nàn, thiết kế không chuyên nghiệp, chức năng không tiện lợi

**- Làm gì để đầu tư cho TMĐT hiệu quả:**

+ Marketing: Một sp mới của DN đc sx ra rồi không có hoạt động marketing nào để quảng bá sản phẩm đó liệu thị trường có biết đến sản phẩm đó hay không

+ Chất lượng website: DN phải đầu tư vào nội dung website bằng cách chăm sóc nội dung thường xuyên, cập nhật thông tin… việc này có thể có 1 nhân viên đứng ra đảm nhiệm

+ Chất lượng dịch vụ hỗ trợ khách hàng: Việc hỗ trợ khách hàng qua mạng 1 cách chuyên nghiệp góp phần rất lớn vào sự thành công của DN trong TMĐT

+ Chiến lược kinh doanh: Nếu DN chọn hình thức kinh doanh nghiêng nhiều về trực tuyến thì nhất định phải có nhân sự chuyên môn về TMĐT như kỹ thuật, marketing qua mạng, hỗ trợ khách hàng qua mạng, nghiên cứu thị trường online…

**- Thương mại điện tử:** là việc thực hiện những hoạt động thương mại dựa trên các công cụ điện tử mà cụ thể là internet. Ví dụ: việc trưng bày hình ảnh hàng hóa, thông tin về doanh nghiệp trên website cũng là 1 phần của TMĐT, liên lạc với khách hàng qua email, tìm kiếm khách hàng đối tác qua việc tìm kiếm thông tin trên internet

Có nhiều cấp độ thực hiện TMĐT. Ở cấp độ cơ bản, DN có thể chỉ mới có website trưng bày thông tin, hình ảnh, tìm kiếm khách hàng qua mạng, liên hệ với khách hàng qua email mà thôi. Cấp độ cao hơn thì DN đã có thể thực hiện 1 số giao dịch trên mạng như cho khách hàng đặt hàng qua mạng, quản lý thông tin khách hàng, đơn đặt hàng bằng CSDL tự động trên mạng, có thể xử lý thanh toán qua mạng bằng thẻ tín dụng…

**- Lợi ích TMĐT:**

+ Quảng bá thông tin, tiếp thị cho thị trường rộng lớn với chi phí cực thấp

+ Dịch vụ tốt hơn cho khách hàng: cung cấp nhiều thông tin, mọi lúc, mọi nơi

+ Với thương mại điện tử: đối tượng khách hàng không còn bị giới hạn về địa lý hay thời gian làm việc, không phải chỉ bán hàng cho cư dân thành phố mà có thể cả VN hoặc thậm chí là nước ngoài, không ngồi chờ khách hàng tìm đến mà có thể tích cự chủ động tìm kiếm khách hàng

+ Với TMĐT không phải tốn kém nhiều cho việc thuê cửa hàng, mặt bằng, đông đảo nhân viên phục vụ…

**- Các bước triển khai:**

+ Xây dựng website

+ Quảng bá website

+ Hỗ trợ khách hàng qua mạng

+ Đổi mới phương thức kinh doanh: đổi mới tư duy trong thời đại @: tốc độ và tính đặc trưng (sự khác biệt)

+ Đổi mới cung cách kinh doanh

**- CSDL là gì:**

+ Một chỗ chứa có tổ chức, tập hợp các dữ liệu có tương quan, các mẫu tin, các cột

+ Hệ quản trị CSDL là CSDL và tập hợp các phần mềm cung cấp các dịch vụ quản trị CSDL cho người lập trình và người dùng cuối.

**- CSDL quan hệ (mô hình quan hệ):** Nền tảng lý thuyết của nó là khái niệm lý thuyết tập hợp trên các quan hệ, tức là tập của các bộ giá trị

+ Thuộc Tính(attribute): Thuộc tính là các đặc điểm riêng của một đối tượng (đối tượng được hiểu như là một loại thực thể ở mô hình thực thể kết hợp), mỗi thuộc tính có một tên gọi và phải thuộc về một kiểu dữ liệu nhất định.

+ Kiểu dữ liệu, Miền giá trị

+ Lược Đồ Quan Hệ: Tập tất cả các thuộc tính cần quản lý của một đối tượng cùng với các mối liên hệ giữa chúng được gọi là lược đồ quan hệ. Nhiều lược đồ quan hệ cùng nằm trong một hệ thống thông tin được gọi là một lược đồ cơ sở dữ liệu.

+ Sự thể hiện của lược đồ quan hệ ở một thời điểm nào đó được gọi là quan hệ, rõ ràng là trên một lược đồ quan hệ có thể xác định nhiều quan hệ. Thường ta dùng các ký hiệu như R,S,Q để chỉ các lược đồ quan hệ, còn quan hệ thường được dùng bởi các ký hiệu là r, s,q

+ Về trực quan thì quan hệ (hay bảng quan hệ) như là một bảng hai chiều gồm các dòng và các cột. Một quan hệ có n thuộc tính được gọi là quan hệ n ngôi. Để chỉ quan hệ r xác định trên lược đồ quan hệ Q ta có thể viết r(Q).

+ Bộ (Tuple): Mẫu tin.

+ Khóa dùng để phân biệt các bộ, một thuộc tính không là khóa của lược đồ quan hệ này nhưng là khóa của lược đồ quan hệ khác gọi là KHÓA NGOẠI.

**- Các phép toán quan hệ:** hợp, giao, trừ, tích đề các, phép chiếu (lấy theo cột), phép chọn (lấy theo dòng), phép kết (join).

**- Ràng buộc toàn vẹn:** not null, ràng buộc khóa chính, ràng buộc khóa ngoại, ràng buộc toàn vẹn miền giá trị

**- Phụ thuộc hàm:** Cho lược đồ quan hệ Q{A1,A2,…,An}. X,Y là hai tập con khác rỗng của Q+. Ta nói X xác định Y (hay Y phụ thuộc hàm vào X) nếu với r là một quan hệ nào đó trên Q, ∀ t1,t2 ∈ r mà t1.X = t2.X ⇒ t1.Y = t2.Y (nghĩa là không thể tồn tại hai bộ trong r giống nhau ở các thuộc tính trong tập X mà lại khác nhau ở một hay nhiều thuộc tính nào đó trong tập Y). Khi đó ta ký hiệu là X → Y. Ví dụ: phụ thuộc hàm của các tập các thuộc tính vào khóa chính

**- Bao đóng của tập thuộc tính:**

Cho lược đồ quan hệ Q(ABCDEGH) và tập phụ thuộc hàm F

F = { f1: B → A; f2: DA → CE; f3: D → H; f4: GH → C; f5: AC → D}

Tìm bao đóng của các tập X = {AC} dựa trên F.

Giải:

X+ (ký hiệu bao đóng) = AC

Do f1, f2, f3, f4 không thoả. f5 thoả :

X+ =ACD

Lập lại bước 2. f1 không thoả, f2 thoả:

X+ =ACDE,

f3 thoả :

X+ =ACDEH

Đến đây rõ ràng không có phụ thuộc hàm nào làm thay đổi X+ nữa, thuật toán dừng lại và kết quả X+ = ACDEH

**- Tìm khóa của lược đồ quan hệ:**

Cho lược đồ quan hệ Q(ABC) và tập phụ thuộc hàm

F={ A→ B; A → C; B → A}

Hãy tìm một khóa của Q.

Giải:

K={A,B,C}

Loại thuộc tính A, do (K-A)+ = Q+ nên K={B,C} thuộc tính B không loại được do (K - B)+ ≠ Q+ nên K={B,C}

Loại thuộc tính C, do (K-C)+ = Q+ nên K={B}.

Vậy một khóa của Q là B.

**- Tìm tất cả các khóa:**

Tìm tất cả các khoá của lược đồ quan hệ Q và tập phụ thuộc hàm F

được cho như sau:

Q(A,B,C);

F={ A→ B; A → C; B → A}

|  |  |  |  |
| --- | --- | --- | --- |
| X | X+ | Siêu khóa | Khóa |
| A | Q+ | A | A |
| B | Q+ | B | B |
| C | C |  |  |
| AB | Q+ | AB |  |
| AC | Q+ | AC |  |
| BC | Q+ | BC |  |
| ABC | Q+ | ABC |  |

Vậy lược đồ quan hệ Q có hai khoá là: {A} và {B}

**- Phủ tối thiếu: (Tập phụ thuộc hàm nhỏ nhất)**

Cho lược đồ quan hệ Q và tập phụ thuộc F như sau:

Q(ABCD) F={ AB→CD; B→C; C→D} Hãy tìm phủ tối thiểu của F.

Giải:

kết quả của bước 1 là: (chuyển vế phải thành 1 thuộc tính)

F={ AB→C; AB→D; B→C; C→D}

kết quả của bước 2 là: (chuyển vế trái thành 1 thuộc tính)

F={ B→C; B→D; B→C; C→D}

kết quả của bước 3 cho phủ tối thiểu: (bỏ phụ thuộc hàm trùng)

F={ B→C; C→D }

**- Dạng chuẩn 1:**

Lược đồ quan hệ Q được gọi là đạt dạng chuẩn 1 (1NF) nếu và chỉ nếu toàn bộ các thuộc tính của Q đều mang giá trị đơn.

Ví dụ:

Ma Ten MonHoc

1 A MH1

MH2

MH3

Không đạt chuẩn 1 🡪 đạt chuẩn 1:

Ma Ten MonHoc

1 A MH1

1 A MH2

1 A MH3

**- Dạng Chuẩn 2 (second normal form)**

Một lược đồ quan hệ Q đạt dạng chuẩn 2 nếu Q đạt dạng chuẩn 1 và tất cả các thuộc tính không khoá của Q đều phụ thuộc đầy đủ vào khoá. Phụ thuộc đầy đủ tức là không có phụ thuộc khác mà vế trái là tập con của phụ thuộc ban đầu

**Dạng Chuẩn 3 (third normal form)**

- Một lược đồ quan hệ Q đạt dạng chuẩn 3 nếu mọi phụ thuộc hàm X→A ∈ F+ ( F là tập phụ thuộc không hiển nhiên định nghĩa trên Q, A là thuộc tính đơn, X là tập thuộc tính con của tập Q+), thì một trong hai điều kiện sau được thoả:

Hoặc X là một siêu khoá của Q

Hoặc A là một thuộc tính khoá

- ĐN2: Lược đồ quan hệ Q ở dạng chuẩn 3 nếu mọi thuộc tính không khóa của Q đều không phụ thuộc bắc cầu vào 1 khóa bất kỳ của Q

- OOP vs struture programming:

+ Hướng cấu trúc xoay quanh cấu trúc dữ liệu và các function. Bussiness Logic đặt hoàn toàn theo các function, cấu trúc dữ liệu chỉ làm vai trò là data container

+ Hướng đối tượng thì ngược lại, chuyển sự tập trung của dev vào dữ liệu tức là thay vì trả lời cho câu hỏi chúng ta muốn **làm gì**, và cần gì để đạt đc thì trả lời câu hỏi chúng ta muốn cái gì, và những thứ đó sẽ làm đc những gì cho ta.

- Lập trình hướng đối tượng: Encapsulation, Inheritance, Polymorphism, Abstraction

xoay quanh đối tượng và dữ liệu chứ không phải thao tác và logic

- Đóng gói: ví dụ hình tam giác thì có 3 đỉnh, tứ giác 4 đỉnh, tính diện tích, không cần phải biết rõ tọa độ các đỉnh đó, chỉ cần gọi phương thức tính diện tích

- Như vậy oop thì dùng lại code, dễ mở rộng, dễ hiểu (việc phân tích bài toán với tư duy mang tính nhân bản, áp dụng từ thực tế vào bài toán), dễ xây dựng và dễ bảo trì

- Ngoài ra oop được chấp nhận và sử dụng rộng rãi, vì thế công cụ tốt, cộng đồng sử dụng đông đảo, open source, tài liệu học, công nghệ mới được xây dựng dựa trên oop…

- Tuy nhiên với chương trình nhỏ thì oop lại cồng kềnh, nặng, chậm

- Java:

+ Java không chỉ là ngôn ngữ, java là 1 nền tảng (platform) với thư viện lớn, cùng với nhiều code dùng lại, và môi trường cung cấp nhiều dịch vụ: security, không phụ thuộc hệ thống, và bộ dọn rác tự động

+ Java simple: dễ hiểu, dễ học: kế thừa từ c++, bỏ những thứ phức tạp, cải tiến thêm chức năng cho dễ dùng

+ Java 🡪 ngôn ngữ lập trình hướng đối tượng

+ Java lập trình mạng rất tốt với sự hỗ trợ từ thư viện lớn, cộng đồng sử dụng lớn và nhiều hỗ trợ từ hãng thứ 3

+ Java có thể chạy trên nhiều platform khác nhau

- Access: phục vụ csdl nhỏ, sql server : csdl lớn, chạy server và phục vụ nhiều người dùng, không có temp table, trigger, store procedure, user-defined functions, views, transaction logging, recovery option: chỉ có simple (last backup), không có recorver to point, không tích hợp với window nt security

<http://www.google.com.vn/url?sa=t&source=web&ct=res&cd=1&ved=0CAkQFjAA&url=http%3A%2F%2Fwww.mssqlcity.com%2FArticles%2FCompare%2Fsql_server_vs_access.htm&rct=j&q=why+sql+server+not+access&ei=4PTbS5baO8GTkAWK9MnNBw&usg=AFQjCNGxEiEdpBESez34qy1N5ly6LEii3Q&sig2=sOc2EOWBab-coISrFk4PJg>