

# **CHƯƠNG 2:**

## **TỔ CHỨC THÔNG TIN TRÊN MÁY TÍNH: BIỂU DIỄN DỮ LIỆU MỨC LOGIC**

Khoa Khoa học và kỹ thuật thông tin  
Bộ môn Thiết bị di động và Công nghệ Web

# Nội dung

1. Mô hình dữ liệu phẳng.
2. Mô hình dữ liệu có cấu trúc.
3. Chỉ mục và Khoá.
4. Mô hình quan hệ.
5. Ràng buộc toàn vẹn.
6. Mô hình dữ liệu XML.

# Mô hình dữ liệu phẳng

# Mô hình dữ liệu phẳng

- Một cơ sở dữ liệu phẳng là một hệ thống cơ sở dữ liệu đơn giản, trong đó mỗi cơ sở dữ liệu được biểu diễn như là một bảng duy nhất.
- Tuy nhiên, một số nhà phát triển ứng dụng vẫn sử dụng các tập tin phẳng để giảm chi phí và tính phức tạp của việc tích hợp cơ sở dữ liệu quan hệ.
- Cơ sở dữ liệu phẳng đôi khi cũng được gọi là cơ sở dữ liệu tệp phẳng (flat-file databases).

# VÍ DỤ

<i>ID No.</i>	<i>Name</i>	<i>D.o.B.</i>	<i>Phone</i>	<i>Class</i>	<i>Tutor</i>	<i>Room</i>
356	Jess	3 Mar 1995	7564356	5B	Mr Noggin	56
412	Hamad	12 Nov 1994	7465846	5B	Mr Noggin	56
459	Sita	9 Jan 1994	8565634	6Y	Ms Take	18
502	Hamad	3 Mar 1995	6554546	5B	Mr Noggin	56

# ƯU VÀ NHƯỢC ĐIỂM CỦA CSDL PHẪNG

## ƯU ĐIỂM

- Nhiều phương thức truy xuất khác nhau:
  - + Tuần tự.
  - + Ngẫu nhiên.
  - + Chỉ mục.

## NHƯỢC ĐIỂM

- Dữ liệu trùng lặp.
- Cần chi phí để xử lý dữ liệu đồng nhất và điều khiển việc truy xuất.
- Bảo mật kém.
- Rất khó để điều khiển việc nhiều người dùng cùng truy cập một lúc.

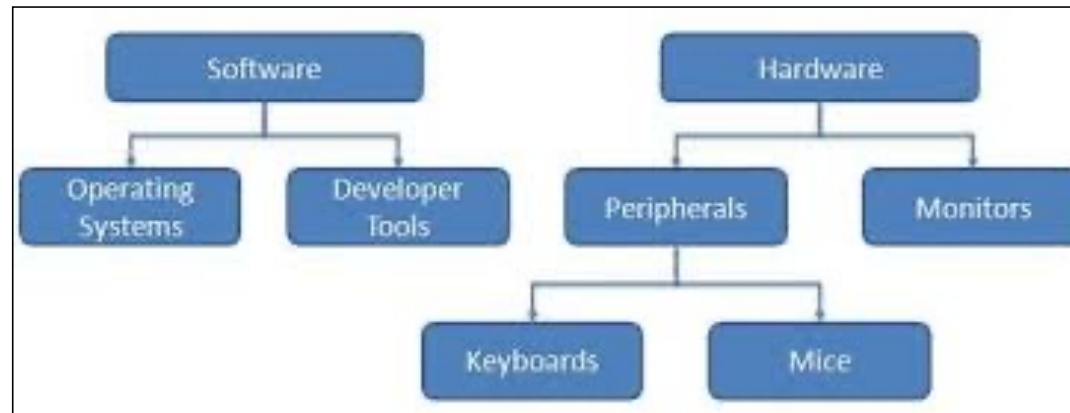
# Mô hình dữ liệu có cấu trúc

# Mô hình phân cấp

- Mô hình phân cấp (Hierarchical model)
  - + Đưa ra vào những năm 60
  - + Dữ liệu được tổ chức thành cấu trúc cây.
  - + Các nút (node) là tập các thực thể.
  - + Các cạnh là các mối quan hệ giữa hai nút theo mối quan hệ nhất định.
- Là mô hình dữ liệu trong đó các bản ghi được sắp xếp theo cấu trúc **top-down (tree)**.
- Một con chỉ có một cha, chỉ có một đường truy nhập tới dữ liệu đó trước.



# Ví dụ MÔ HÌNH PHÂN CẤP



# ƯU VÀ NHƯỢC ĐIỂM CỦA MÔ HÌNH PHÂN CẤP

## ƯU ĐIỂM

- Đơn giản về khái niệm:
  - + Nhóm dữ liệu có thể được liên quan đến nhau.
  - + Dữ liệu liên quan có thể được xem với nhau.
- Tập trung dữ liệu.
- Giảm sự dư thừa và phát huy tính nhất quán.

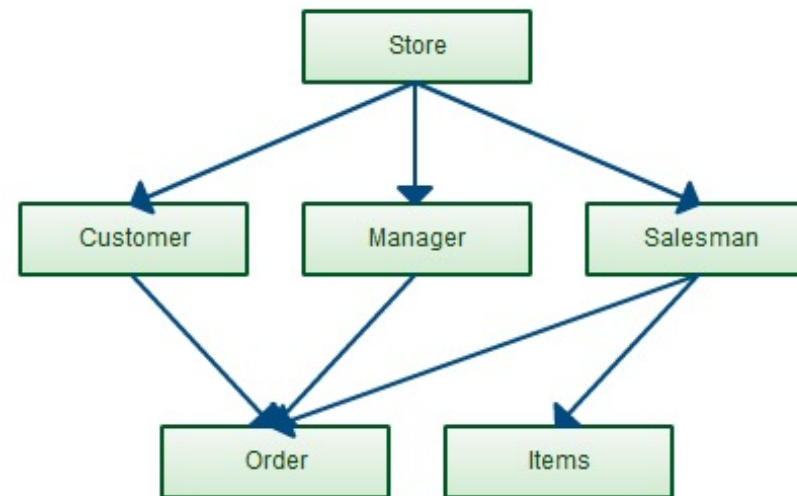
## NHƯỢC ĐIỂM

- Biểu diễn hạn chế về mối quan hệ dữ liệu:
  - + không cho phép quan hệ(M: N)
- Thực hiện phức tạp:
  - + cần kiến thức chuyên sâu về lưu trữ dữ liệu vật lý
- Sự phụ thuộc cấu trúc:
  - + truy cập dữ liệu đòi hỏi đường dẫn lưu trữ vật lý

# Mô hình mạng

- Được đưa vào cuối những năm 60.
- Dữ liệu được tổ chức thành một đồ thị có hướng.
- Các đỉnh là các thực thể.
- Các cung là quan hệ giữa hai đỉnh, một kiểu bản ghi có thể liên kết với nhiều kiểu bản ghi khác.
- Một con có thể có nhiều cha, có nhiều đường truy nhập đến một dữ liệu cho trước.

# Ví dụ MÔ HÌNH MẠNG



# ƯU VÀ NHƯỢC ĐIỂM CỦA MÔ HÌNH MẠNG

## ƯU ĐIỂM

- Cho phép nhiều loại quan hệ dữ liệu.
- Hiệu quả và truy cập dữ liệu linh hoạt.
- Phù hợp với các tiêu chuẩn.
- Tăng cường quản trị cơ sở dữ liệu.

## NHƯỢC ĐIỂM

- Hệ thống phức tạp
- Đòi hỏi phải có sự quen thuộc với cấu trúc nội bộ để truy cập dữ liệu.
- Thay đổi cấu trúc nhỏ đòi hỏi thay đổi chương trình.

# Chỉ mục và Khoá

# Chỉ mục

- Chỉ mục (index) là **bảng tra cứu đặc biệt** mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thu thập dữ liệu.
- Phân loại:
  - + Chỉ mục trên cột là khóa.
  - + Chỉ mục trên cột không là khóa.
  - + **B Tree.**

# Ví dụ về chỉ mục trong thực tế

## Index

### A

*addictions* 35, 39  
*the Arctic* 68, 69

### B

*the bar* 48, 68, 69, 75, 81, 89, 90, 132, 149  
*bush tips* 36, 38, 52, 57, 58, 65, 66, 79, 129  
*business* 34, 39, 48, 76, 77, 116, 128, 132, 145, 156

### C

*city life* 41, 54, 72, 74, 98, 108, 116, 135, 146, 147

### D

*dog's life* 12, 16, 21, 22, 24, 25, 32, 40, 43, 53, 57, 61, 62, 66, 71, 75, 86, 87, 89, 91, 92, 95, 96, 98, 100, 101, 102, 103, 112, 115, 125, 126, 129, 144, 152, 160

### E

*the environment* 26, 33, 62, 77, 78, 82, 85, 121, 124, 128, 134, 136, 137, 138, 139, 147, 148, 149, 150, 151, 152, 160, 163  
*explorers* 17, 28, 29, 77, 114

### F

*fall* 13, 83  
*fishing* 6, 9, 10, 19, 46, 78, 127, 157, 159

*flying* 22, 23

### G

*government* 12, 17, 25, 43, 49, 57, 63, 68, 70, 76, 79, 80, 99, 114, 126, 127, 133, 140, 143, 146, 151, 153, 155

*grouse* 24, 27, 163

*guiding* 17, 61, 86, 122, 128, 136, 158

### H

*health* 30  
*hippies* 15, 35, 36, 37, 88  
*horses* 47, 122, 144

*hunting* 14, 31, 32, 46, 61, 67, 70, 72, 85, 86, 106, 107, 124, 127, 141, 151, 161, 162

### K

*kids* 24, 30, 58, 59, 74, 123, 137

### M

*marriage* 23, 33, 36, 49, 52, 56, 60, 66, 68, 71, 75, 76, 89, 90, 96, 97, 100, 102, 124, 141, 154, 157  
*mechanics* 18, 66, 87, 145  
*money* 32

*mushing* 1, 2, 3, 4, 5, 9, 10, 18, 20, 21, 34, 37, 43, 53, 55, 73, 74, 75, 89, 94, 98, 109, 112, 150

### O

*oldtimers* 43, 93, 107, 126, 156, 158

### P

*philosophy* 94, 102, 108, 109, 113, 115, 116, 121, 123, 130, 132, 134, 142, 144, 146, 152, 154, 155, 156, 157, 159, 161, 162, 164

*police* 11, 13, 27, 34, 44

*politics* 6, 44, 56, 67, 71, 73, 125, 139, 140, 142

*prospecting* 18, 29, 31, 64, 104, 123, 130

*punks* 16, 19, 47, 108, 130

### R

*ravens* 21, 30, 47, 67, 71, 95, 104, 124, 134, 137, 138, 139, 140, 165  
*religion* 7, 38, 45, 58  
*romance* 4, 5, 16, 23, 35, 42, 50, 52, 55, 56, 71, 114, 145

### S

*scientists* 11, 25, 28, 38, 63, 65, 67, 90, 120, 122, 127, 135, 137, 138, 139, 140, 150, 155, 163, 164  
*snowmachines* 20, 149

*southerners* 49, 60, 62, 63, 72, 74, 76, 80, 81, 82, 99, 101, 103, 113, 120, 122, 131, 133, 136, 142, 146, 152, 153, 154, 158, 159, 160  
*summer* 10, 27, 45, 64, 157

### T

*tourism* 8, 21, 26, 33, 41, 44, 45, 46, 64, 80, 81, 83, 120, 125, 133, 135, 143, 158

*tradition* 84, 98

*trapping* 14, 19, 32, 56, 60, 72, 92, 94, 97, 107, 109, 112, 113, 128, 153, 165

### W

*weather* 20, 50, 93, 101, 115  
*winter* 50, 51, 107

Philosophy, 145, 170, 204, 239, 249, 276-277, 285, 287.

Greek, 149, 238.

Plagues, 57-70, 81-82.

Plato, 21, 226.

Platonic Archetype, 109, 139, 140.

Forms, 226, 229, 240, 275, 285, 288.

in the Tabernacle, 208, 226, 228-229, 232-233, 240, 242, 244, 257, 266.

Politics, 35, 36, 143, 192, 232. See also Tyranny.

Polytheism, 121-122, 124-125, 214.

See also Interfaith, Idolotry.

Priestly Garments, 251-258.

Priests, 76, 84, 144, 167, 212, 225-258 passim, 259, 262, 265, 272. See also High Priest.

Revelation, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000.

Saadiah Gaon, 123, 215.

Sabbath, 9, 14, 104-105, 122, 131-134, 196-197, 267-268, 287-288.

Sabbatical Year, 196

Sacrifice, 42, 44, 51-52, 74, 159, 160, 162, 167, 189, 197-198, 200-201, 256, 259, 262, 269.

Blemished, 259.

Burnt Offering, 210-211, 247, 261. of Firstborn and First Fruit, 84-85, 144 182, 211-212, 286.

Grain Offering, 247.

Incense, 231-232, 236-237, 252-253, 263-265.

Insincere or Impious Offerings, 15, 227-228, 246-247, 290.

of Paschal Lamb, 74-79, 80.

Peace Offering, 210-211.

in Tabernacle or Temple, 225-226,



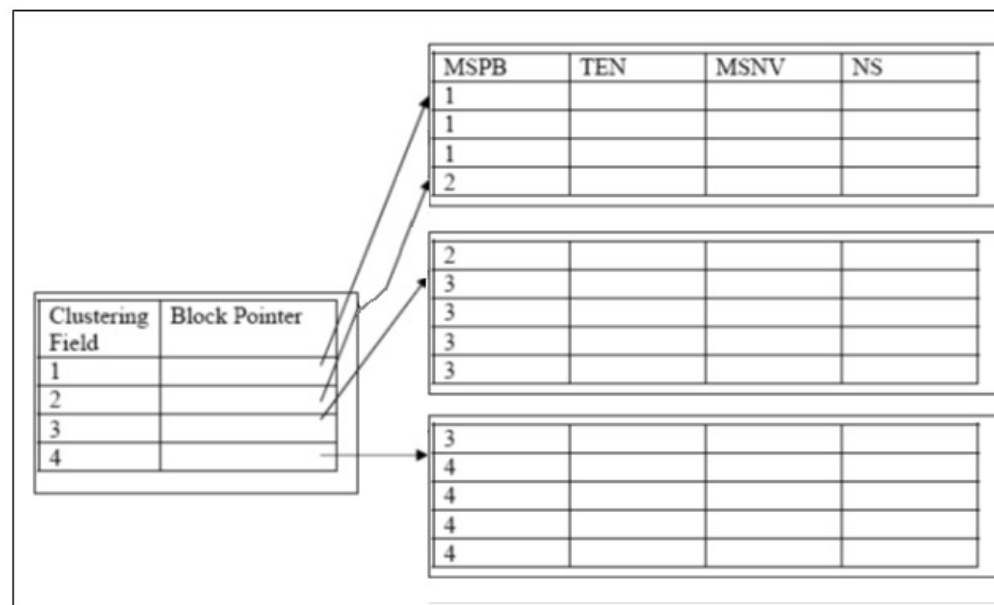
# Chỉ mục tên cột là khoá

- Tập tin chỉ mục là một FILE với các mẫu tin có 2 cột: khóa và địa chỉ block, đã được sắp trên thuộc tính khóa. Địa chỉ khóa cho biết vị trí của block chứa mẫu tin trên đĩa. Tập tin chỉ mục có kích thước nhỏ hơn nhiều so với FILE dữ liệu chính. Vì vậy tập tin chỉ mục sẽ được đọc vào bộ nhớ chính khi chương trình CSDL khởi động.
- Vì tập tin chỉ mục đã được sắp nên nó dùng thuật toán tìm kiếm nhị phân khi tìm kiếm.
- Nếu 1 mẫu tin trong tập tin chỉ mục tương ứng 1 mẫu tin trong File dữ liệu chính ta gọi chỉ mục dày (dense index). Nếu 1 mẫu tin trong tập tin chỉ mục tương ứng nhiều mẫu tin trong File dữ liệu chính ta gọi chỉ mục thưa (nondense index)

# Chỉ mục trên cột không là khoá

- Nếu các mẫu tin đã được sắp trên một cột không phải khóa (key) và có giá trị có thể lặp lại, cột đó được gọi là cột Clustering (clustering field), trường hợp này ta tạo ra một file chỉ mục gọi là Clustering Index.
- Clustering Index có 2 cột: cột 1 giống với cột clustering của file dữ liệu chính, cột thứ 2 chứa con trỏ khối chỉ tới đầu khối có chứa dữ liệu cột clustering. Giá trị trên cột 1 trong Clustering Index là duy nhất. Nó là 1 ví dụ của tập tin chỉ mục thưa.

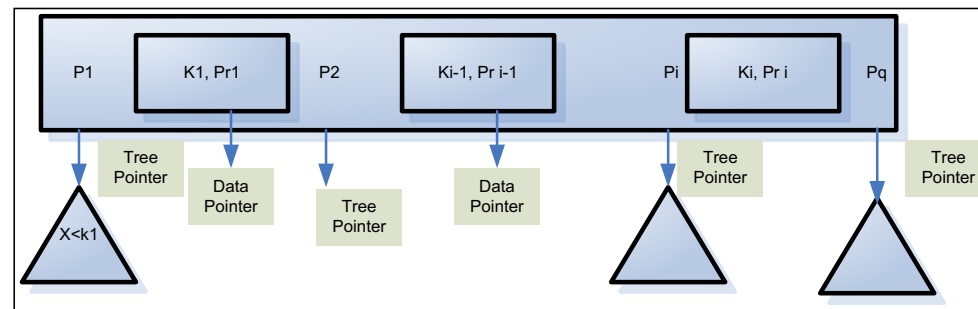
# Ví dụ



# B-Tree

- Định nghĩa: **B Tree** là cây tìm kiếm nhiều nhánh thỏa điều kiện sau
  - + Tất cả các node lá có cùng độ cao.
  - + Tất cả các Node trung gian có nhiều nhất  $m$  cây con và có ít nhất  $m/2$  cây con khác rỗng.
  - + Node gốc có thể có  $m$  cây con hay có thể có 2 cây con không phải là lá.
  - + Các giá trị khóa trên các Node đã sắp (tăng dần).
  - + Gọi  $k$  là số cây con của một Node thì số khóa nằm trên Node là  $k-1$ .
  - + Các Node có cùng cấu trúc dữ liệu.
  - + Mỗi Node có cấu trúc như sau :  **$\langle P1, \langle K1, Q1 \rangle, P2, \langle K2, Q2 \rangle, \dots \langle \dots \rangle \rangle$**
- Trong đó :
  - +  **$P_i$**  là con trỏ đến một Node khác trong cây
  - +  **$K_i$**  là khóa
  - +  **$Q_i$**  là con trỏ dữ liệu, trỏ đến block trên FILE dữ liệu chứa mẫu tin có khóa là  $K_i$ .

# Ví dụ B-Tree



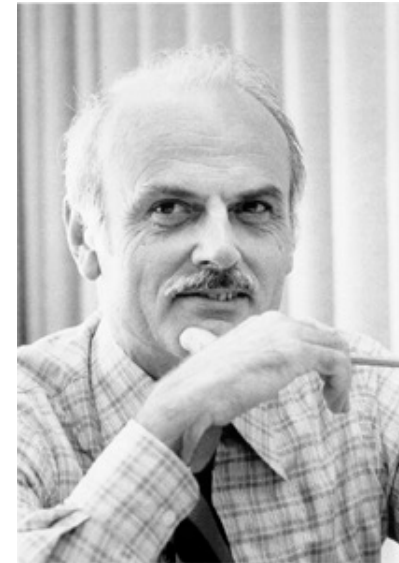
# Đánh giá về B-Tree

- Vì có nhiều khóa trên một Node và có nhiều Node trên cùng độ cao nên việc tìm kiếm trên B Tree khá tốt. Càng nhiều mẫu tin trên một Node thì cây có độ cao càng nhỏ nhằm cải tiến tốc độ truy xuất đĩa.
- Tổ chức theo BTree không những nhanh hơn so với File có thứ tự mà việc thêm, xóa cũng hiệu quả hơn.

# Mô hình quan hệ

# Giới thiệu

- E.F Codd đưa vào đầu những năm 70.
- Dựa trên lý thuyết tập hợp và đại số quan hệ.
- Vì tính chất chặt chẽ của toán học về lý thuyết tập hợp nên mô hình này đã mô tả dữ liệu một cách rõ ràng, mềm dẻo và là mô hình thông dụng.
- Nhiều hệ QTCSDL đều tổ chức dữ liệu theo mô hình dữ liệu quan hệ.
- Trong đó dữ liệu được tổ chức dưới dạng bảng, các phép toán thao tác trên dữ liệu dựa trên lý thuyết tập hợp của toán học.



Edgar F Codd (1923-2003)

<https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>



# Các thành phần chính của mô hình quan hệ

- Các khái niệm chính:
  - + Quan hệ - ứng với bảng.
  - + Bộ - ứng với dòng.
  - + Thuộc tính – ứng với cột.
- Tân từ.
- Thể hiện của một quan hệ
- Lược đồ quan hệ.

# Quan hệ

- Các thông tin lưu trữ trong CSDL được tổ chức thành bảng (table) gọi là quan hệ.

NHANVIEN						
MaNV	HoNV	TenNV	Email	SDT	Phong	Luong
NV01	Nguyen Van	A	email1@domain.com	0123456789	1	20000
NV02	Tran Thanh	B	email2@domain.com	0987643210	1	22000
NV03	Tran Thi	C	email3@domain.org	0388888888	3	15000

# Bộ

- Bộ là **các dòng** của quan hệ (trừ dòng tiêu đề: tên của các thuộc tính).
- Thể hiện dữ liệu cụ thể của các thuộc tính trong quan hệ.

NHANVIEN						
MaNV	HoNV	TenNV	Email	SDT	Phong	Luong
NV01	Nguyen Van	A	email1@domain.com	0123456789	1	20000
NV02	Tran Thanh	B	email2@domain.com	0987643210	1	22000
NV03	Tran Thi	C	email3@domain.org	0388888888	3	15000

# Thuộc tính

- Thuộc tính:
  - + **Tên gọi:** dãy ký tự (gợi nhớ)
  - + **Kiểu dữ liệu:** Số, Chuỗi, Thời gian, Luận lý, OLE.
  - + **Miền giá trị:** tập giá trị mà thuộc tính có thể nhận.
- Ký hiệu miền giá trị của thuộc tính A là  $Dom(A)$ .
- Một thuộc tính không có giá trị hoặc chưa xác định được giá trị => **giá trị Null.**

# Thuộc tính

— Tên các cột của quan hệ:

+ Mô tả ý nghĩa cho các giá trị tại cột đó.

+ Tất cả các dữ liệu trong cùng một cột đều có cùng kiểu dữ liệu.

NHANVIEN						
<i>MaNV</i>	<i>HoNV</i>	<i>TenNV</i>	<i>Email</i>	<i>SDT</i>	<i>Phong</i>	<i>Luong</i>
NV01	Nguyen Van	A	email1@domain.com	0123456789	1	20000
NV02	Tran Thanh	B	email2@domain.com	0987643210	1	22000
NV03	Tran Thi	C	email3@domain.org	0388888888	3	15000

# Thể hiện của quan hệ

- Thể hiện của một quan hệ là tập hợp các bộ giá trị của quan hệ tại một thời điểm nhất định.
- VD: Thể hiện cho quan hệ Nhân viên gồm các bộ (dòng) như sau:

NHANVIEN						
<i>MaNV</i>	<i>HoNV</i>	<i>TenNV</i>	<i>Email</i>	<i>SDT</i>	<i>Phong</i>	<i>Luong</i>
NV01	Nguyen Van	A	email1@domain.com	0123456789	1	20000
NV02	Tran Thanh	B	email2@domain.com	0987643210	1	22000
NV03	Tran Thi	C	email3@domain.org	0388888888	3	15000

# Lược đồ quan hệ

- Cấu trúc của một quan hệ: là tập thuộc tính hình thành nên quan hệ.
- Một lược đồ quan hệ gồm:
  - + Một tập thuộc tính của quan hệ.
  - + Một mô tả để xác định ý nghĩa và mối liên hệ giữa các thuộc tính.
- Lược đồ quan hệ được đặc trưng bởi:
  - + Một tên phân biệt.
  - + Một tập hợp hữu hạn các thuộc tính  $(A1, ..., An)$ .
- Ký hiệu:  $Q(A1, A2, ..., An)$ .

## Ví dụ: quan hệ nhân viên

- Tên quan hệ: NhanVien.
- Các thuộc tính: MaNV, HoNV, TenNV, Email, SoDT, Phong, Luong.
- Ký hiệu:

**NhanVien(MaNV, HoNV, TenNV, Email, SoDT, Phong, Luong)**



# Tên từ

- Định nghĩa: Tên từ là một quy tắc dùng để mô tả một quan hệ.
- Ký hiệu:  $||Q||$ .
- Ví dụ:

NhanVien(MaNV, HoNV, TenNV, Email, SoDT, Phong, Luong)

$||\text{NhanVien}||$ : MaNV – mã nhân viên; HoNV – họ nhân viên; TenNV – tên nhân viên; Email – email của nhân viên; SoDT - số điện thoại nhân viên; Phong – Phòng làm việc của nhân viên; Luong – lương của nhân viên.

# Các phép toán trên quan hệ

- Phép chiếu:  $\sigma$
- Phép chọn:  $\Pi$
- Phép hội:  $\cup$
- Phép giao:  $\cap$
- Phép trừ:  $\setminus$
- Phép chia:  $\div$
- Phép kết:  $\bowtie$
- Phép gom nhóm:  $\Join$

# Ràng buộc dữ liệu

- Là những quy tắc, điều kiện, ràng buộc cần thỏa mãn cho mọi dữ liệu, để dữ liệu đúng đắn và thống nhất.
- Ràng buộc toàn vẹn (RBTV-Integrity Constraint) là các quy tắc, ràng buộc lên cơ sở dữ liệu, để hạn chế tình trạng xấu của cơ sở dữ liệu, không phản ánh đúng thế giới mà nó đang biểu diễn.
- Các quy tắc có thể xuất phát từ:
  - + Bản thân mô hình dữ liệu.
  - + Quy tắc trong quản lý.
  - + Quy tắc trong tự nhiên.

## Ví dụ

NhanVien (MSNV, NgaySinh, NV Cty)

Ta có:

$\text{NgaySinh} < \text{NV Cty}$ .

Giải thích: Ngày vào công ty của nhân viên phải lớn hơn ngày sinh của nhân viên

# Các loại ràng buộc

- Ràng buộc miền giá trị (value range).
- Ràng buộc liên thuộc tính một quan hệ (Many fields-one relation)
  - + Là ràng buộc xảy ra trên một quan hệ và nhiều thuộc tính.
- Ràng buộc liên thuộc tính nhiều quan hệ (Many fields-Many relations)
  - + Là ràng buộc xảy ra trên nhiều quan hệ và nhiều thuộc tính (không kể ràng buộc khoá ngoại)
- Ràng buộc toàn vẹn do xuất hiện chu trình.
- Khoá chính (primary key).
- Khoá ngoại (foreign key).

# Ràng buộc miền giá trị

KETQUA (MSMH, MSSV, DIEM, NAMHOC, HOCKI...)

Ta có các ràng buộc:

$0 \leq DIEM \leq 10$

$HOCKI = \{1, 2\}$

Ví dụ sai:

MSSV	MSMH	DIEM	NAMHOC	HOCKI
1	CSDL	1A	2013-2014	1
1	MMT	9	2013-2014	4
2	CSDL	5	2013-2014	2

# Ràng buộc liên thuộc tính một quan hệ

CTHD (MSHD, MSMH, SL, DG, TTIEEN)

Ràng buộc:  $SL \times DG = TTIEEN$ .

Ví dụ sai:

MSHD	MSMH	SL	DG	TTIEEN
1	2	10	10	100
1	3	20	25	500
2	1	13	20	265

# Ràng buộc liên thuộc tính nhiều quan hệ

HOADON (MSHD, THANHTIEN)

CTHD (MSHD, MSMH, SL, DG, TTIEEN)

Ràng buộc:

Với mỗi HD:  $THANHTIEN = SL \times DG$

Với mỗi CTHD:  $TTIEN = SL \times DG$

Ví dụ sai:

MSHD	THANHTIEN
1	1000
2	200
3	500

MSHD	MSMH	SL	DG	TTIEN
1	1	10	20	200
1	2	20	40	800
2	1	20	10	200
3	2	10	10	100
3	3	20	30	600



# Ràng buộc toàn vẹn do xuất hiện chu trình

LOPHOC(MSLOP, MSKHOA, TENLOP)

HOCSINH(MSHS, MSLOP, MSKHOA)

MSLOP	MSKHOA	TENLOP
TH01	TH	..
TH02	TH	..
DT01	DT	..

MSHS	MSLOP	MSKHOA
TH97	TH01	TH
<i>DT90</i>	<i>DT01</i>	<i>TH</i>
TH98	TH02	TH

# Ràng buộc khoá chính

- **Toàn vẹn thực thể:** là một trong những ba quy tắc toàn vẹn vốn có. Toàn vẹn thực thể là một nguyên tắc toàn vẹn trong đó nói rằng mỗi bảng phải có một khóa chính và các cột được chọn làm khóa chính phải là duy nhất và không NULL.
- **Khóa chính:** Trong cơ sở dữ liệu quan hệ, một khóa chính là một tập hợp một hoặc nhiều thuộc tính, giá trị trong PK bảo đảm là duy nhất cho mỗi tuple (hàng) trong một mối quan hệ

# Ví dụ sai về PK

<u>MSSV</u>	<u>MSMH</u>	<u>LAN</u>	KHOA	DIEM
1	1	1	2002	10
2	1	2	2002	3
2	1	1	2002	3
1	1	1	2002	9

<u>MSSV</u>	<u>MSMH</u>	<u>LAN</u>	KHOA	DIEM
1	1		2002	10
2	1	2	2002	3
2	1	1	2002	3

<u>MSSV</u>	TENSV	NAMSINH	QUEQUAN
1	A	1/2/2002	TPHCM
2	B	2/3/2002	CANTHO
1	1	4/5/2001	DONGNAI
1	1	4/5/2001	TPHCM

# Ràng buộc khoá ngoại (tham chiếu)

- Xét R1 và R2 là 2 lược đồ, X được gọi là khoá ngoại của R1 tham chiếu đến R2 nếu thoả:
  - + x là khoá chính của R2. Giá trị trên R2 phải tồn tại trước R1.
  - + Khoá chính và khoá ngoại phải cùng kiểu dữ liệu, cùng miền giá trị. (Đặc biệt R1 có thể LÀ R2).

# Đặc điểm khoá ngoại

- Có thể NULL.
- Một lược đồ quan hệ có thể có nhiều khoá ngoại.
- Một thuộc tính có thể vừa tham gia khoá ngoại vừa tham gia khoá chính.
- Tên khoá chính và khoá ngoại có thể khác nhau.
- Khoá chính và khoá ngoại có thể nằm trên một lược đồ.

## Ví dụ

### — Ví dụ 1 (sai).

- + SINHVIEN(MSSV text(10), TENSX, QUEQUAN, MSLOP,...)
- + KETQUA(MSSV int, MSMH, DIEM)

### — Ví dụ 2: Cho 3 quan hệ

- + Nhân viên (MSNV, TenNV,.....) (i)
  - + DeAn\_NhanVien (MSNV, MSDA, TGian) )(ii)
  - + DeAn (MSDA, TenDA) (iii)
- Ta nói: MSDA trong (ii) là khoá ngoại tham chiếu đến MSDA (iii)
- MSNV trong (ii) là khoá ngoại tham chiếu MSNV(i)

## Ví dụ (tt)

### — Ví dụ 3:

- + SINHVIEN (MSSV, TENS<sub>V</sub>, QUEQUAN, MSLOP,...).
- + KETQUA (MSSV, MSMH, DIEM).
- + MSSV trong KETQUA vừa tham gia khoá ngoại và khoá chính.

### — Ví dụ 4:

- + NHANKHAU (MSNK, TENNK, TONGIAO,.....).
- + PHIEUKETHON (MSPKH, MSNK1, MSNK2,...).
- + Tên khoá chính và khoá ngoại có thể khác.
- + PHIEUKETHON có 2 khoá ngoại.

# Các lưu ý

- Khi thêm xoá sửa dữ liệu có thể vi phạm các ràng buộc PK và FK
- Thứ tự tạo và xoá bảng (table) có ý nghĩa
- Thứ tự tạo và xoá dữ liệu trong bảng (table) có ý nghĩa



# Một số trường hợp đặc biệt

- Trường hợp 1:
  - + R1 (A, B, C, D)
  - + R2 (A, C, M)
  - + Kết quả: Không có khóa ngoại
- Trường hợp 2:
  - + R1 (A, B, C, D)
  - + R2 (K, C, A, B)
  - + Kết quả: có khoá ngoại là bộ (A, B)

# Ví dụ về vi phạm khoá ngoại

**Ví dụ 3.14** Sửa vi phạm khoá ngoại  
NHANVIEN

MSNV	TENNV	QUEQUAN	MSPB
1	A	XX	12
2	B	YY	2
3	C	ZZ	1
4	A	VV	1

PHONGBAN

MSPB	TENPB
1	BB
2	CC
3	DD

**Ví dụ 3.15** Xoá vi phạm khoá ngoại

NHANVIEN

MSNV	TENNV	QUEQUAN	MSPB
1	A	XX	12
2	B	YY	2
3	C	ZZ	1
4	A	VV	1

PHONGBAN

MSPB	TENPB
1	BB
2	CC
3	DD

Các chú ý khác:

# MÔ HÌNH DỮ LIỆU XML

# GIỚI THIỆU

- Cơ sở dữ liệu nhỏ.
- Nền tảng: Win32, WinCE, PalmOS, Linux, Solaris
- Tìm kiếm, thêm, cập nhật và xóa dữ liệu trong cơ sở dữ liệu được tạo ra bởi các tập tin XML.
- Các cơ sở dữ liệu có thể là một phần của một ứng dụng.

# XML

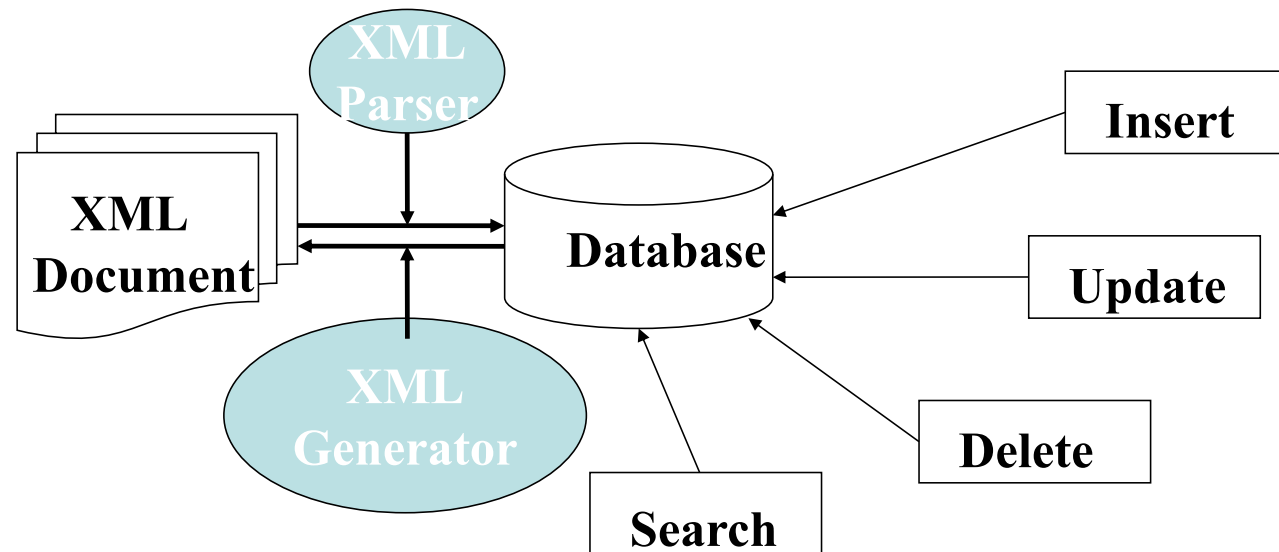
- Là một dạng ngôn ngữ đánh dấu (markup language).
- Được sử dụng để tạo ra cấu trúc cho dữ liệu.
- Gồm 2 thành phần chính:
  - + Thẻ:
    - Thẻ mở đầu: ký hiệu <tag>.
    - Thẻ kết thúc: ký hiệu </tag>.
    - Thuộc tính cho thẻ (nếu có).
  - + Nội dung.

# Ví dụ

```
<Telephone>
  <EntryID>1038</EntryID>
  <LoginName>jake</LoginName>
  <PassWord>pass38</PassWord>
  <Lastname>Kim</Lastname>
  <Firstname>Jungkee</Firstname>
  <Date_of_Birth>10.01.1964</Date_of_Birth>
  <Company>FSU</Company>
  <Salutation>Mr.</Salutation>
  <Email>jake@csit.fsu.edu</Email>
  <Address><Street>400 Dirac Science Library</Street>
    <City>Tallahassee</City>
    <ZIP>32306</ZIP>
    <Country>Korea</Country>
    <Telephone>6447018</Telephone>
    <Fax />
  </Address>
</Telephone>
```

```
<university>
  <department>
    <dept_name> Comp. Sci. </dept_name>
    <building> Taylor </building>
    <budget> 100000 </budget>
  </department>
  <course>
    <course_id> CS-101 </course_id>
    <title> Intro. to Computer Science </title>
    <dept_name> Comp. Sci </dept_name>
    <credits> 4 </credits>
  </course>
</university>
```

# Cấu trúc cơ bản của một CSDL XML



Để truy vấn dữ liệu trên CSDL XML, ta dùng Xpath và Xquery

# TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Ty, *Cơ sở dữ liệu*, NXB Đại học quốc gia TP HCM (2010).
3. Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hùng, *Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TP HCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.
5. E. Codd, *A relational model of data for large shared data banks*, Communications of the ACM (1970).





# Bài tập

- Tạo 1 CSDL mẫu
  - + Phát hiện các ràng buộc trên CSDL
  - + Cài đặt CSDL và các ràng buộc này trên SQL Server
- Tạo 1 CSDL mẫu
  - + Thêm vào các bảng dữ liệu mẫu bằng vòng lặp for (khoảng 100000 dòng).
  - + Viết câu select đơn giản trên 1 cột (c1) chưa index và đề ý đến thời gian thực hiện T1
  - + Chọn các cột trên các quan hệ để index
  - + Cài đặt các index trên SQL Server
  - + Viết câu select đơn giản trên 1 cột (c1) đã index và so sánh thời gian thực hiện T2 với T1

# PHỤ LỤC: CHUẨN HOÁ CSDL

# Giới thiệu

- Thiết kế cơ sở dữ liệu là 1 phần cực kì quan trọng, nếu thiết kế cẩn thận thì sau này sẽ tiết kiệm được rất nhiều thời gian trong quá trình phát triển. Và để tối ưu cơ sở dữ liệu thì nên tuân theo các chuẩn thiết kế.
- Có 4 loại dạng chuẩn như sau:
  - + **First Normal Form (1NF)**: dạng chuẩn 1NF
  - + **Second Normal Form (2NF)**: dạng chuẩn 2NF
  - + **Third Normal Form (3NF)**: dạng chuẩn 3NF
  - + **Boyce-Codd Normal Form (BCNF)**: dạng chuẩn Boyce-Codd

# Dạng chuẩn 1

— Điều kiện:

- + *Lược đồ quan hệ R ở dạng chuẩn 1 (1NF-First Normal Form) nếu mọi thuộc tính của R đều chứa các giá trị nguyên tố (atomic value), giá trị này không là một danh sách các giá trị hoặc giá trị phức hợp (composite value)*

## Ví dụ

— Các thuộc tính chưa là giá trị nguyên tố:

+ **TENMON** → *CSDLAnh* có thể phân rã ra thành: CSDL và Anh.

MASV	HOTEN	DIACHI	MAMON	TENMON	DIEM
A01	Lê Na	12 Thái Hà	M01M02	CSDLAnh	89
A02	Trần An	56 Mã Mây	M01	CSDL	8
A03	Hà Nam	24 Cầu Gỗ	M01M02M03	CSDLAnhToán 1	689

## Dạng chuẩn 2

— Điều kiện:

- + *Lược đồ quan hệ  $R$  ở dạng chuẩn 2 (2NF-Second Normal Form) đối với tập phụ thuộc hàm  $F$  nếu  $R$  ở dạng chuẩn 1 và mọi thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào mọi khóa của  $R$ .*

# Phụ thuộc hàm

- Cho một lược đồ quan hệ  $R(U)$ ,  $r$  là một quan hệ bất kỳ trên lược đồ quan hệ  $R$ ,  $X$  và  $Y$  là hai tập thuộc tính con của  $U$ . Phụ thuộc hàm (FD-Functional Dependency)  $X \rightarrow Y$  trên lược đồ quan hệ  $R$ , được đọc là " $X$  xác định hàm  $Y$ " hoặc " $y$  phụ thuộc hàm vào  $X$ ", nếu:

$$\forall t_1, t_2 \in r(R): t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

- Tức là mỗi giá trị của  $X$  trong  $r$  chỉ tương ứng với một giá trị của  $Y$ .



## Dạng chuẩn 3

— Điều kiện:

- + *Phải đạt chuẩn 2NF.*
- + *Mọi thuộc tính không khóa phụ thuộc bắc cầu vào thuộc tính khóa (nghĩa là tất cả các thuộc tính không khóa phải được suy ra trực tiếp từ thuộc tính khóa).*

# Dạng chuẩn 4 (Boyce-Codd-Kent)

— Điều kiện:

+ *Phải đạt chuẩn 3NF.*

+ *Không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa.*