**Nhóm 3**

Câu hỏi ôn tập chương 2.

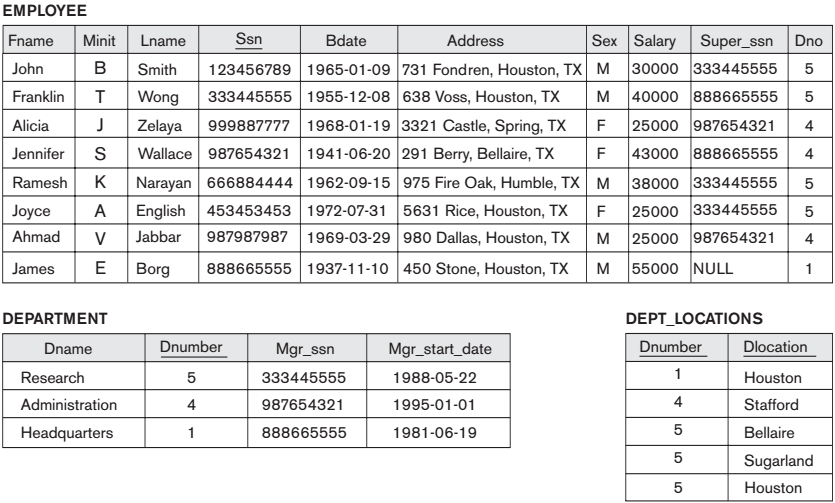
1. Quan hệ là gì? =>một quan hệ là một bảng chứa dữ liệu bao gồm nhiều cột và nhiều dòng. Dữ liệu trong một quan hệ phải thỏa các ràng buộc liên quan tới quan hệ đó.
2. Mỗi cột trong một quan hệ còn được gọi là gì? => một thuộc tính (attribute) hay một trường (field) của quan hệ.
3. Mỗi dòng trong một quan hệ còn được gọi là gì? => một bộ (tuple) hay mẫu tin/ bản ghi (record).
4. Ràng buộc toàn vẹn là gì?=> là tập các quy tắc mà mọi dữ liệu trong CSDL phải tuân theo.
5. Các quan hệ phải có ràng buộc toàn vẹn để làm gì?=> nhằm đảm bảo tính toàn vẹn của cơ sở dữ liệu.
6. Miền giá trị (MGT) của một thuộc tính là gì? => là một tập giá trị mà một thuộc tính có thể nhận.
7. Lược đồ quan hệ là gì?=> là tập tất cả các thuộc tính có trong quan hệ.
8. Biểu diễn nào dưới đây là lược đồ quan hệ => a
9. Q(A, B, C)
10. Q 🡨 A, B, C
11. Q{A, B, C}
12. Cả 3 câu trên đều đúng
13. Thể hiện/ tình trạng của một quan hệ là gì? => là tập tất cả các bộ của một quan hệ tại một thời điểm.
14. Cơ sở dữ liệu (CSDL) quan hệ là gì?=> là một tập các quan hệ có liên quan với nhau về một lĩnh vực cụ thể.
15. Lược đồ cơ sở dữ liệu (CSDL) quan hệ là gì?=> là một tập các lược đồ quan hệ có liên quan với nhau về một lĩnh vực cụ thể.
16. Siêu khóa của một lược đồ quan hệ Q(R) là gì?=> *k* được gọi là siêu khóa nếu các giá trị của k có thể xác định duy nhất một bộ trong quan hệ.
17. Trong một quan hệ Chỉ có duy nhất 1 siêu khóa là đúng hay sai=> sai.
18. Một siêu khóa trong một quan hệ có thể bao gồm nhiều hơn số thuộc tính trong một quan hệ là đúng hay sai? => sai
19. Phát biểu nào dưới đây sai: => sai
20. Số thuộc tính trong siêu khóa là tập con của tập thuộc tính trong quan hệ
21. Tập cha của siêu khóa cũng chính là siêu khóa
22. Một tập con bất kỳ của siêu khóa cũng là siêu khóa
23. Cả 3 câu trên đều đúng
24. Cho Q(A, B, C, D, E) có siêu khóa là {A, B, C}. Phát biểu nào dưới đây là sai: => c
25. {A, B, C, D} là siêu khóa
26. {A, B, C, D, E} là siêu khóa
27. Cả 2 câu a, b đều đúng
28. Cả 2 câu a, b đều sai
29. Khóa ứng viên (candidate key) của một quan hệ là gì? => Siêu khóa *k* được gọi là **khóa ứng viên** nếu k là tập nhỏ nhất.
30. Phát biểu nào sau đây đúng => a
31. Khóa ứng viên là khóa có thể được chọn làm khóa chính cho một quan hệ
32. Mỗi quan hệ chỉ có duy nhất một khóa ứng viên
33. Cả a và b đều đúng
34. Cả a và b đều sai
35. Khóa chính (primary key) của một quan hệ là gì? => là một trong các khóa ứng viên sẽ được chọn làm **khóa chính** cho quan hệ.
36. Mỗi quan hệ chỉ có duy nhất 1 khóa chính là đúng hay sai? => sai, một quan hệ có thể có nhiều khoá chính, nhưng chỉ nên có 1 khoá chính trong 1 quan hệ.
37. Cho Q1(A, B, C) và Q2(D, A). Phát biểu nào dưới đây là đúng =>d
38. Thuộc tính Q2.A là khóa ngoại trong Q2
39. Q1 được gọi là quan hệ được tham chiếu và Q2 được gọi là quan hệ tham chiếu
40. Thuộc tính Q2.A tham chiếu qua Q1.A
41. Cả 3 câu trên đều đúng
42. Cho Q1(A, B, C) và Q2(D, B). Phát biểu nào dưới đây là đúng => c
43. Q2.B có thể là khóa ngoại tham chiếu qua Q1.B nếu Q1.B có ràng buộc unique
44. Q2.B là khóa ngoại tham chiếu qua Q1.B nếu Q1.B có ràng buộc not null
45. Cả 2 câu a, b đều đúng
46. Cả 2 câu a, b đều đúng
47. Phát biểu nào dưới đây đúng => c
48. Khóa ngoại trong mô hình quan hệ biểu diễn mối kết hợp giữa 2 quan hệ
49. Ràng buộc về khóa ngoại còn gọi là ràng buộc tham chiếu
50. Cả 2 câu a, b đều đúng
51. Cả 2 câu a, b đều đúng
52. Giải thích ràng buộc tham chiếu nghĩa giữa Q1(A, B, C) và Q2(D, A)  
    =>Q2.A là khoá ngoại tham chiếu qua Q1.A
53. Một thuộc tính có ràng buộc unique nghĩa là gì? => mỗi giá trị trong cột có ràng buộc này phải là duy nhất trong cột đó.
54. Một thuộc tính có ràng buộc not null nghĩa là gì? => Các ô trong cột có ràng buộc này bắt buộc phải có giá trị khác null
55. Một thuộc tính có ràng buộc khóa chính nghĩa là gì?=> các giá trị là khóa chính phải là unique và not null.
56. Miền giá trị của một thuộc tính nghĩa có phải là một ràng buộc áp đặt lên dữ liệu của thuộc tính đó không? => Phải. Các giá trị trong cột phải nằm trong miền giá trị của cột.
57. Các ký hiệu sau |Sex|, Dom(Sex), MGT(Sex) ký hiệu nào là ký hiệu miền giá trị của thuộc tính Sex. => |Sex|, Dom(Sex) là ký hiệu miền giá trị của thuộc tính Sex.
58. Cho NV(MaNV, Hoten, NgaySinh) và PB(MaPB, TenPB, Matrph). Quy định: mỗi phòng ban có 1 NV là trưởng phòng. Phát biểu nào dưới đây là đúng 2 lược đồ quan hệ trên =>d
59. MaNV có ràng buộc unique và not null
60. Có ràng buộc tham chiếu từ PB.Matrph sang NV.MaNV
61. Cả a và b đều sai
62. Cả a và b đều đúng
63. Trong một quan hệ, thứ tự các thuộc tính và các bộ có quan trọng không? Tại sao? => Không vì nó không bị ràng buộc thể thứ tự , và chúng ta vẫn có thể chọn bất kì để làm khóa chính cho quan hệ đó và đánh dấu bằng nét liền gạch dưới.

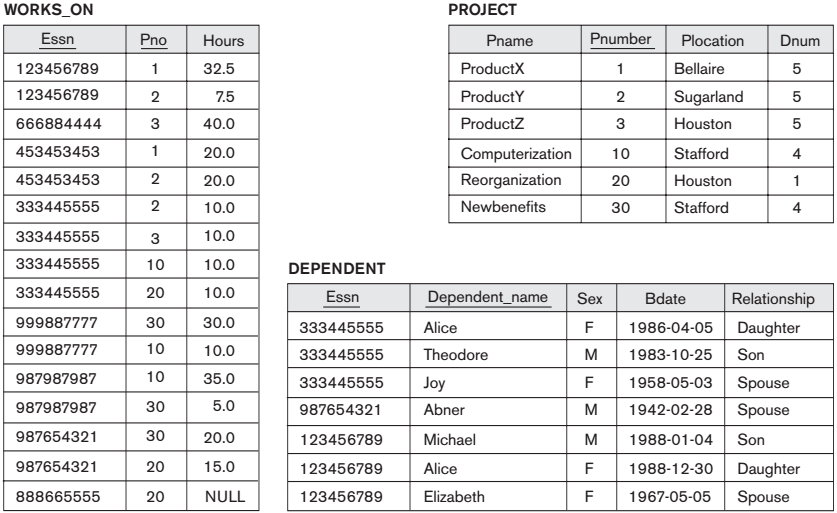
Review questions

1. Describe the three-schema architecture.  
   => Physical level: describes how a record (e.g., customer) is stored.  
   Logical level: describes data stored in database, and the relationships among the data. View level: application programs hide details of data types. Views can also hide information (such as an employee’s salary) for security purposes.
2. Define the following terms and give an example for each term: an attribute, the domain of an attribute, a relation schema, a relation, n-tuple, degree of a relation, a relationship, a relation instance (state), a relational database schema, a relational database state, integrity constraints. :  
   =>   
   +) attribute: are the describing characteristics or properties that define all items pertaining to a certain category applied to all cells of a column. Eg: SinhVien have attribute such as: mssv, hoTen,…  
     
   +) the domain of an attribute: are rules that describe the legal values of a field type. Eg: Dom(Sex) = {‘Male’,’Female’}  
     
   +) a relation schema: is the set of all attributes contained in the relation. Eg: SINHVIEN(MaSV,HoTen,NgaySinh).  
     
   +) n-tuple: An n-tuple is a sequence (or ordered list) of n elements, where n is a non-negative integer: Eg. 1 table has 4 tuple so if we order 3 tuple in this relation that is called 3-tuple  
     
   +) degree of a relation: represents the number of entity types that associate in a relationship. Eg: 1-1, 1-n, n-n.  
     
   +) a relationship: associations between tables that are created using join statements to retrieve data. Eg: SINHVIEN(MSSV, HoTen, NgaySinh), KetQua(MSSV, MaMH, Diem).  
     
   +) a relation instance: The set of all tuples of a relation at a time.  
     
   +) a relational database schema: a set of related schemas about a particular domain. Eg: SINHVIEN(MaSV, HoTen, NgaySinh, Diachi), MONHOC(MaMH, TenMH, SoTC), KQTHI(MaSV, MaMH, Diem).  
     
   +) a relational database state: A database is always in one specific state. For example, these states include ONLINE, OFFLINE, or SUSPECT.  
     
   +) integrity constraints: A set of rules that all data in a database must follow to ensure database integrity. Eg: unique, not null, Foreign key constraint,…
3. Why are tuples in a relation not ordered? => because a relation attempts to represent facts at a logical or abstract level.
4. Why are duplicate tuples not allowed in a relation? => because they create redundancy of database that makes the data processing like inserting, deleting, querying, updating, etc. which reduces the speed of the database.
5. What is the difference between a key and a superkey? => a super key is an attribute (or set of attributes) that is used to uniquely identifies all attributes in a relation. A key is an attribute.
6. Discuss the entity integrity and referential integrity constraints. Why is each considered important? => ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.

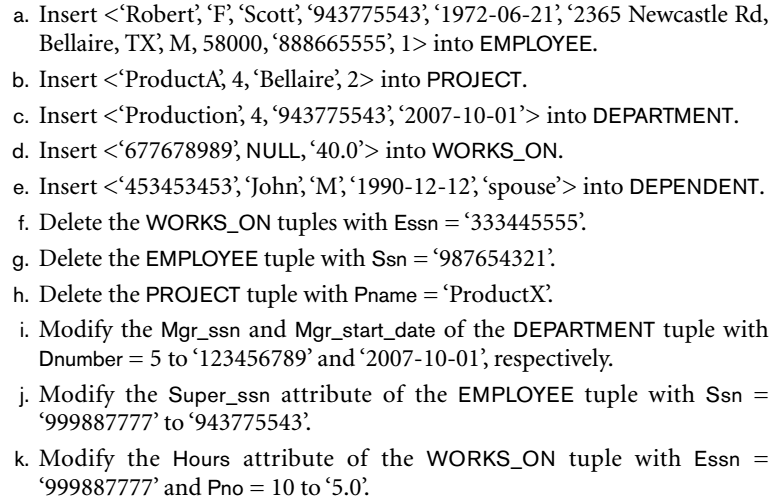
Exercises.

1. Given one possible database state for the COMPANY relational database schema as follows:





- Suppose that each of the following Update operations is applied directly to the database state shown above. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.



Answer:

Câu lệnh trên thực hiện chèn 1 dòng dữ liệu vào trong bảng Employee với dữ liệu

tương ứng với các cột từ trái sang phải (như ‘Robert’ nằm trong cột Fname, …)

Ta thấy bảng này có các ràng buộc toàn vẹn (RBTV):

- về miền giá trị (kiểu dữ liệu): Tất cả dữ liệu trong câu lệnh đều thỏa RBTV này

tương ứng trong các cột.

- Ràng buộc khóa ngoại (còn gọi là ràng buộc tham chiếu):

1. cột Super\_ssn là khóa ngoại tham chiếu đến cột Ssn (i.e. người lãnh đạo

cũng chính là nhân viên của công ty. Ví dụ: nhân viên John B. Smith dưới

quyền lãnh đạo của người có mã ‘333445555’ là người có tên Franklin T.

Wong). Trong câu lệnh Insert ở trên, người lãnh đạo có mã ‘888665555’.

Người này đã có trong bảng. Vậy ràng buộc này thỏa.

2. cột Dno là khóa ngoại, tham chiếu đến cột Dnumber của bảng Department.

Trong câu lệnh Insert trên, giá trị cột Dno được chèn vào là 1. Và cột này đã

có bên bảng Department. Như vậy, ràng buộc này thỏa.

Kết luận: Câu lệnh trên có thể thực hiện mà không vi phàm các RBTV.

Các RBTV liên quan tới 2 bảng này:

- về miền giá trị (kiểu dữ liệu) đối với bảng PROJECT: Tất cả dữ liệu trong câu lệnh đều thỏa RBTV này tương ứng trong các cột.

- RB về khóa ngoại: Cột Dnum trong bảng PROJECT là khóa ngoại tham chiếu qua cột Dnumber bên bảng DEPARTMENT. Trong câu lệnh trên, giá trị cột Dnum là 2, nhưng tìm trong cột Dnumber trong bảng DEPARTMENT ta không thấy giá trị này.

Như vậy câu lệnh trên không thực hiện được vì vi phạm ràng buộc này. Muốn thực hiện được thao tác của câu lệnh này ta phải thực hiện chèn thêm 1 department (phòng ban) có Dnumber (mã phòng ban) là 2 vào trong bảng DEPARTMENT, sau đó ta mới thực hiện được câu lệnh trên.

1. 

Về miền giá trị(kiểu dữ liệu) với bảng WORKS\_ON ta thấy giá trị chèn vào cột thỏa tất cả dữ liệu các cột.

Về ràng buộc khóa chính: Essn là khóa ngoại trong bảng WORKS\_ON nên khi muốn thêm một tuple ta cần kiểm tra giá trị Essn này đã xuất hiện ở quan hệ được tham chiếu hay chưa. Ở trường hợp này Essn = 677678989 trong WORKS\_ON tham chiếu đến Ssn trong EMPLOYEE và giá trị Ssn này không tồn tại trong EMPLOYEE.  
 Như vậy không thể thực hiện được câu lệnh, muốn thực hiện thao tác trên ta cần phải đổi một giá trị ở cột Ssn trong bảng EMPLOYEE thành 677678989 hoặc thêm một tuple mới có trị Ssn = 677678989 rồi mới thực hiện lệnh insert trên.

Về miền giá trị (kiểu dữ liệu) với bảng DEPENDENT ta thấy giá trị chèn vào thỏa RBTV tương ứng với các cột của bảng.

Về ràng buộc khóa chính : Ta thấy cột Essn của bảng DEPENDENT là khóa chính tham chiếu qua cột Essn của bảng WORKS\_ON và ta thấy có tồn tại giá trị 453453453 trong vột Essn của bảng WORKS\_ON. Vậy ràng buộc này thỏa.

Như vậy thao tác trên được thực hiện.

Không vi phạm vì các thuộc tính Essn, Pno trong bảng WORKS\_ON là khoá ngoại tham chiếu tương ứng tới cột Ssn của bảng EMPLOYEE và cột Pnumber của bảng PROJECT; và vì các thuộc tính của bảng WORKS\_ON đều không phải là thuộc tính được tham chiếu. Nếu xoá các tuple thuộc bảng WORKS\_ON sẽ không ảnh hưởng.

=> Về miền giá trị (kiểu dữ liệu) với bảng DEPENDENT ta thấy giá trị chèn vào thỏa RBTV tương ứng với các cột của bảng.

=> Vi phạm ràng buộc khóa ngoại, vì Ssn trong bảng EMPLOYEE được tham chiếu bởi cột Essn trong 2 bảng WORKS\_ON và DEPENDENT. Trong 2 bảng này, giá trị Essn = ‘987654321’ đều tồn tại và tham chiếu đến Ssn = ‘987654321’ trong bảng EMPLOYEE.

=> Muốn thực hiện câu lệnh này mà không vi phạm ràng buộc toàn vẹn thì thay đổi giá trị của cột Essn của các tuple có giá trị là ‘987654321’ trong 2 bảng WORKS\_ON và DEPENDENT, hoặc ta cũng có thể xóa các tuple vi phạm này đi. Sau đó ta có thể thực hiện câu lệnh delete trên.

=> Về ràng buộc khoá ngoại: cột Pno thuộc bảng WORKS\_ON tham chiếu tới cột Pnumber thuộc bảng PROJECT. Vì thế khi xoá tuple có Pname là ProductX thì các tuple có Pname = ‘ProductX’ trong bảng PROJECT sẽ bị xóa, mà trong trường hợp này tuple có Pname = ‘ProductX’ lại có Pnumber = 1, khi xóa giá trị này các tuple thuộc bảng WORKS\_ON có cột Pno có giá trị là 1 sẽ không tham chiếu được cột Pnumber thuộc bảng PROJECT nữa.

=> Muốn thực hiện được câu lệnh này mà không vi phạm đến các ràng buộc toàn vẹn thì trong bảng WORKS\_ON, ta cần thay đổi các giá trị Pno của các tuple có Pno = 1 hoặc ta có thể xóa các tuple này đi sau đó mới thực hiện câu lệnh delete trên.

Có thể biến đổi được :

Về kiểu dữ liệu phù hợp với cột Mgr\_ssn và Mgr\_start\_date trong bảng DEPARTMENT.

Về khóa ngoại : ta thấy giá trị 123456789 của cột Mgr\_ssn tham chiếu qua cột Ssn của bảng EMPLOYEE có tồn tại giá trị trên. Và giá trị Mgr\_start\_date không bị ràng buộc .

Như vậy câu lênh trên có thể thực hiện được.

Cột Super\_ssn là khóa ngoại tham chiếu đến cột Ssn (i.e. người lãnh đạo cũng chính là nhân viên của công ty. Trong câu lệnh này, giá trị cột Super\_Ssn bị thay thế thành “943775543” không phù hợp. Vì cột Ssn không tồn tại giá trị nào là “943775543”.

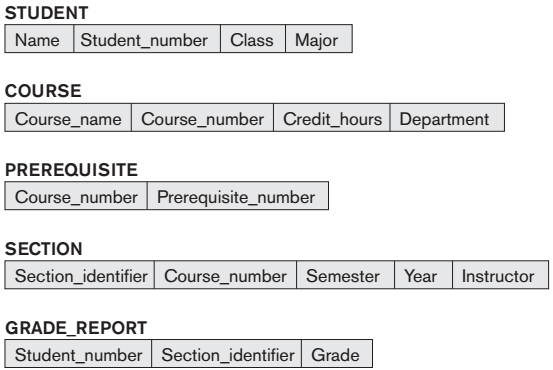
Muốn thực hiện được thao tác, ta đổi lại giá trị cột Ssn của một tuple trong EMPLOYEE lại thành “943775543” hoặc thêm mới một tuple có giá trị Ssn = “943775543”, sau đó mới thực hiện lệnh modify trên.

Về giá trị : giá trị cột Hours thay thế thõa mãn RBTV.

Về ràng buộc tham chiếu: Hours không tham chiếu đến bảng nào.

=>Câu lệnh có thể thực hiện được

1. Given a University database as follows:



Consider the above figure. In addition to constraints relating the values of columns in one table to columns in another table, there are also constraints that impose restrictions on values in a column or a combination of columns within a table. One such constraint dictates that a column or a group of columns must be unique across all rows in the table. For example, in the STUDENT table, the Student\_number column must be unique (to prevent two different students from having the same Student\_number).

- Identify the column or the group of columns in the other tables that must be unique across all rows in the table.

- Identify the primary key, foreign key in each relational schema.

Answer:

STUDENT:

- Primary key: Student\_number

- Unique: (Student\_number), (Student\_number, Name), (Student\_number, Class), (Student\_number, Major), (Student\_number, Name, Class), (Student\_number, Name, Major), (Student\_number, Class, Major), (Course\_name, Credit\_hours, Department)

COURSE:

- Primary key: Course\_number

- Unique: (Course\_number), (Course\_number, Course\_name), (Course\_number, Credit\_hour), (Course\_number, Department), (Course\_number, Course\_name, Credit\_hours), (Course\_number, Course\_name, Department), (Course\_number, Credit\_hours, Department), (Course\_name, Credit\_hours, Department)

PREREQUISITE:

- Primary key: {Prerequisite\_number, Course\_number}

- Foreign key: Course\_number

- Unique: (Prerequisite\_number, Course\_number)

SECTION:

- Primary key: Section\_identifier

- Foreign key: Course\_number

- Unique: (Section\_identifier)

GRADE\_REPORT

- Primary key: {Student\_number, Section\_identifier}

- Foreign key: Student\_number, Section\_identifier

- Unique: (Student\_number, Section\_identifier)

- Specify the following queries on the University database schema:

a. Retrieve the student\_number and the name of all students in the university.

b. Retrieve the student\_number and the name of all students in “Computer Science” major

c. Retrieve the prerequisite course number and course name of the course which has course number “1111’’

d. Retrieve the section identifier, course number and course name of courses which are opened in semester 1, year 2019

e. Retrieve the section identifier, course number and course name of courses which are not opened in semester 1, year 2019

f. Retrieve the section identifier, course number and course name of courses which are opened in year 2018 or 2019.

g. Retrieve the section identifier, course number and course name of courses which are opened in year 2018 and 2019.

Answer:

1. Π Student\_number, Name(STUDENT)
2. Π Student\_number, Name(σMajor = ‘*Computer Science’*(STUDENT))
3. Π Course\_name, Course\_number(σCourse\_number = *‘1111’*(COURSE)) |X| PREREQUISITE
4. Π Course\_name, Course\_number, Section\_identifier (σsemester = 1 and year = 2019 (SECTION IXI SOURSE))
5. Π Section\_identifier, Course\_number, Course\_name(SECTION |X| COURSE - σSemester = 1 and Year = *2019*(SECTION |X| COURSE))
6. Π Section\_identifier, Course\_number(σYear = 2018 or Year = *2019*(SECTION)) |X| Π Course\_name, Course\_number (COURSE)
7. Π Section\_identifier, Course\_number, Course\_name(σYear = 2018(SECTION) ∩ σYear = *2019*(SECTION) |X| COURSE)