Chapter 7

Integrity Constraints
(Ràng buộc toàn vẹn)

TRUONG DAI HOC KHOA HOC TU NHIEN
TP. HO CHI MINH
KHTN

cdio 4.0

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Content

- **Overview**
- Characteristics of Integrity Constraints
- Classification
- Implementation

Overview

- Integrity Constraints (IC) are discovered from the semantics of the data or the representation of the data in business[1].
- IC is to ensure [1]:
  - The correctness of the data and data schema.
  - The semantic of the database schema
- When an IC is declared, all instances of a database must satisfy the IC at any time [1].
- An IC is discovered and declared by database designer during the design phase [1].
- An IC is defined on a relational schema or involves in may relational schemas [1].

[1] *Fundamental of Databases 4th , Ramez Elmasri & Shamkant B. Navathe, ISBN 0-321-12226-7, 2003*

Content

- Overview
- **Characteristics of Integrity Constraints**
  - Context
  - Content
  - Table of influence
- Classification
- Implementation

Context

- Context of an IC
  - Relational schemas that are likely to be violated by the IC when performing data updates (adding, deleting, modifying data)

- Example (IC_1)
  - The salary of a lecturer cannot exceed the head of the department
    - Update operations that affect to the IC1
      - Update the salary of the lecturer
      - Add new a lecturer to department
      - Appoint a new department head
    - Context: GIAOVIEN, BOMON

Context (cont.)

- Example (IC_2)
  - The direct manager (of a lecturer) must be a lecturer in the same department
    - Update operations that affect the IC2
      - Update line manager of a lecturer
      - Add new a lecturer
    - Context: GIAOVIEN

Content

- The content of a IC is presented by
  - Natural language
    - Easy to understand but lack of formal, consistency
  - Formal language
    - Concise, consistent but sometimes difficult to understand
    - Language forms
      - Relational Algebra
      - Tuples Relational Calculus
      - Pseudo code (mã giả)

Content (cont.)

- Example (IC_1)
  - Natural language
    - Mức lương của một người giáo viên không được vượt quá trưởng bộ môn của giáo viên đó.
  - Formal language

$$
\begin{array}{l}
(\forall t)(\text{GIAOVIEN}(t) \land (\exists s)(\text{BOMON}(s) \land \\
\qquad\qquad\qquad\qquad (\exists u)(\text{GIAOVIEN}(u) \land \\
\qquad\qquad\qquad\qquad\qquad s.\text{TRUONGBM} = u.\text{MAGV} \land \\
\qquad\qquad\qquad\qquad\qquad s.\text{MABM} = t.\text{MABM} \land \\
\qquad\qquad\qquad\qquad\qquad t.\text{LUONG} \leq u.\text{LUONG})))
\end{array}
$$

Content (cont.)

- Example (IC_2)
  - Natural language
    - Người quản lý trực tiếp của một giáo viên phải là một giáo viên trong cùng bộ môn
  - Formal language

$$
\begin{aligned}
&(\forall t)(\mathrm{GIAOVIEN}(t) \wedge (t.\mathrm{GVQLCM} \neq \mathrm{null} \;\Rightarrow \\
&\qquad (\exists s)(\mathrm{GIAOVIEN}(s) \wedge s.\mathrm{MABM} = t.\mathrm{MABM} \\
&\qquad\qquad \wedge\, s.\mathrm{MAGV} = t.\mathrm{GVQLCM})))
\end{aligned}
$$

Influence table

- Determine which update operations need to check for IC when performed on the relations context

- 2 types
  - Influence table for 1 IC
  - Aggregate influence table for all ICs

Influence table for 1 IC

IC title	Insert	Delete	Update
Relation 1	+	–	+ (Attribute)
Relation 2	–	+	–
…			
Relation n	–	+	–

(+) Violation of IC
(–) Not violation of IC

Aggregate influence table

	IC 1			IC 2						IC m		
	T	X	S	T	X	S	…	…	…	T	X	S
Relation 1	+	-	+	+	-	+				+	-	+
Relation 2	-	+	-									
Relation 3	-	-	+							-	+	-
…												
Relation n				-	+	-				-	-	+

Content

- Overview
- Characteristics of Integrity Constraints
- **Classification**
- Implementation

IC Classification

- ICs are classified into three main categories [1]:
  - Inherent model based constraints (RBTV bắt buộc liên quan đến mô hình dữ liệu).
    - Ex: A relation cannot contain duplicate tuples.
  - Schema based constraints (RBTV liên quan đến lược đồ của mô hình dữ liệu).
    - Ex: Value domain constraints, key constraints, null constraints, reference constraints.
  - Application based constraints (RBTV dựa trên ứng dụng).
    - Ex: The salary of a lecturer cannot exceed the department head

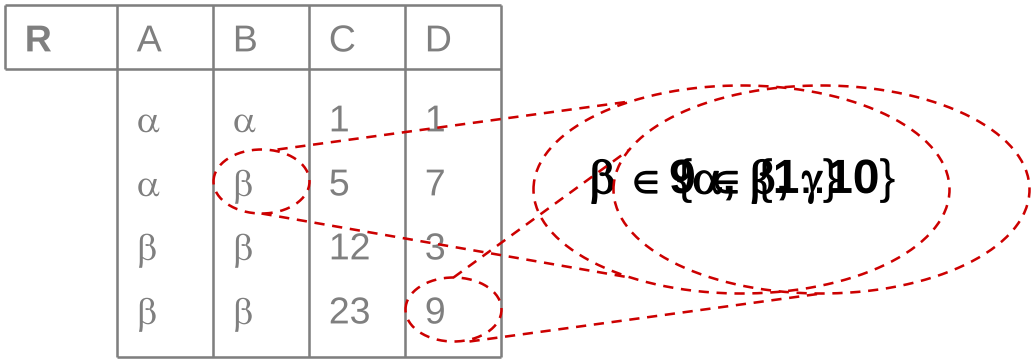
[1] *Fundamental of Databases 4th , Ramez Elmasri & Shamkant B. Navathe, ISBN 0-321-12226-7, 2003*

IC Classification

- IC having the context of one relation:
  - Domain value
  - Multiple tuples
  - Multiple attributes
- IC having the context of many relation :
  - Reference (tham chiếu)
  - Multiple tuples through multiple relations (liên bộ liên quan hệ)
  - Multiple attributes through multiple relations (liên thuộc tính liên quan hệ)
  - Inference/calculated attribute (thuộc tính suy diễn)
  - Relations form a cycle in relation graph (chu trình)

IC domain value

- Constraints specifying values for an attribute



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

- Domain value
  - Continuous (liên tục)
  - Disjointed (rời rạc)

Example 3

- Lecturer's gender must be ‘Nam’ or ‘Nữ’
  - Context: GIÁOVIÊN
  - Content:

$$(\forall t)(GIAOVIEN(t) \wedge (t.PHAI = \text{'Nam'} \vee t.PHAI = \text{‘Nữ‘}))$$

or

$$DOM(PHAI) = \{\text{‘Nam’}, \text{‘Nữ’}\}$$

  - Influence table:

IC3	Insert	Delete	Update
GIÁOVIÊN	+	–	+ (PHAI)

Example 4

- Allowance for each work in project must not exceed 20 million VND.

  - Context: THAMGIAĐT

  - Content:

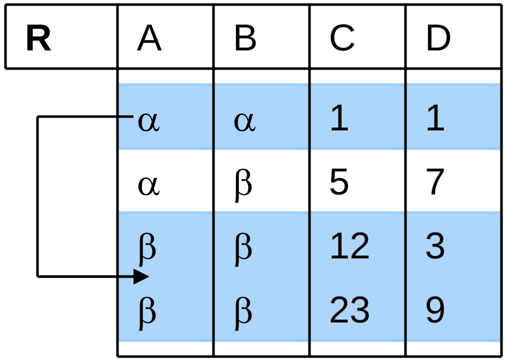
    $(\forall t)(\text{THAMGIAĐT}(t) \wedge t.\text{PHỤCẤP} \leq 20)$

  - Influence table:

IC4	Insert	Delete	Update
THAMGIAĐT	+	–	+ (PHỤCẤP)

IC - Multiple tuples

- The existence of one or more tuples depends on the existence of one or more other tuples in the same relation



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

- Special cases
  - IC – primary key
  - IC unique (key)

Example 5

- Department name is unique.
  - Context: BOMON
  - Content:

$$\boxed{\begin{array}{l}(\forall t1, t2)\ (BOMON(t1) \wedge BOMON(t2) \wedge \\ \qquad\qquad (t1 \neq t2 \Rightarrow t1.TENBM \neq t2.TENBM))\end{array}}$$

or

$$\boxed{\begin{array}{l}(\forall t1)(BOMON(t1) \wedge \neg(\exists t2)(BOMON(t2) \wedge \\ \qquad\qquad t1 \neq t2 \wedge t1.TENBM = t2.TENBM)\ )\end{array}}$$

  - Influence table:

IC_5	Insert	Delete	Update
BOMON	+	–	+ (TENBM)

Example 6

- A lecturer can participate in up to 5 works in all projects

  - Context: THAMGIAĐT

  - Content:

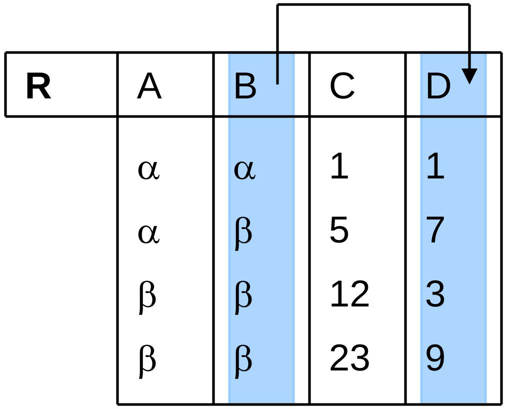
    $$(\forall t)(\text{THAMGIADT}(t) \land \text{card}(\{ s \mid \text{THAMGIADT}(s) \land s.\text{MAGV} = t.\text{MAGV}\}) \leq 5)$$

  - Influence table:

IC6	Insert	Delete	Update
THAMGIADT	+	–	+ (MAGV)

IC – Multiple attributes

- Constraint between attributes in the same relation



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

Example 8

- A lecturer must not manage him/herself (Một giáo viên không trực tiếp quản lý chuyên môn chính mình)
  - Context: GIAOVIEN
  - Content:

$$(\forall t)(\text{GIAOVIEN}(t) \wedge (t.\text{GVQLCM} = \text{null} \vee t.\text{GVQLCM} \neq t.\text{MAGV}) \}$$

  - Influence table:

IC8	Insert	Delete	Update
GIAOVIEN	+	–	+ (GVQLCM)

Example 9

- The start date of a project is always smaller than the end date.

  - Context: ĐỀTÀI

  - Content:

    $(\forall t)(\text{ĐỀTÀI}(t) \wedge t.\text{NGÀYBĐ} \leq t.\text{NGÀYKT})$

  - Influence table:

IC9	Insert	Delete	Update
ĐỀTÀI	+	–	+ (NGÀYBĐ, NGÀYKT)

Example 10

- The start date of a work is always smaller than the end date of that work.
  - Context: CÔNGVIỆC
  - Content:

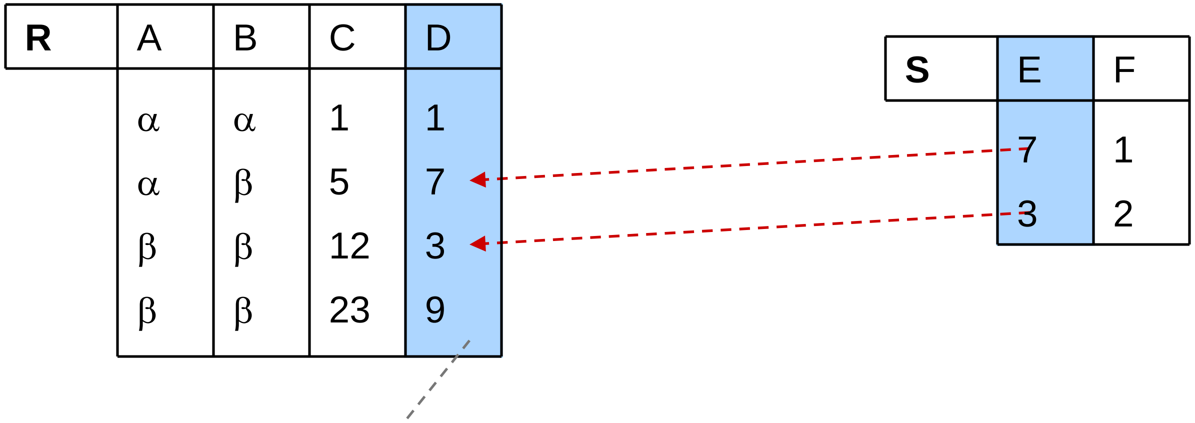
$$(\forall t)(\text{ CÔNGVIỆC}(t) \wedge t.\text{NGÀYBĐ} \ \leq \ t.\text{NGÀYKT})$$

  - Influence table:

IC_{10}	Insert	Delete	Update
CÔNGVIỆC	+	–	+ (NGÀYBĐ, NGÀYKT)

IC - Reference

- Values of attributes in a relation must refer to the values of attributes in other relation.



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

S	E	F
	7	1
	3	2

Bắt buộc phải tồn tại trước

- Special case
  - IC foreign key

Example 11

- Every lecturer must belong to/work in a department.

  - Context: BOMON, GIAOVIEN
  - Content:

$$(\forall t)(GIAOVIEN(t) \wedge (\exists s)(BOMON(s) \wedge s.MABM = t.MABM))$$

  - Influence table:

IC11	Insert	Delete	Update
GIAOVIEN	+	-	+ (MABM)
BOMON	-	+	+(MABM)

Example 12

- Department head must be a lecturer
  - Context: BOMON, GIAOVIEN
  - Content:

$$(\forall t)(BOMON(t) \wedge (\exists s)(GIAOVIEN(s) \wedge s.MAGV = t.TRUONGBM))$$

  - Influence table:

IC_{12}	Insert	Delete	Update
GIAOVIEN	-	+	+ (MAGV)
BOMON	+	-	+(TRUONGBM)

IC - Reference (cont.)

- Also called existence dependency
- The context is two relations, but there are cases where it degenerates into a relation
  - Example (IC_2)
    - The line manager of a lecturer must be a lecturer in the same department
    - Context: GIAOVIEN
    - Content:

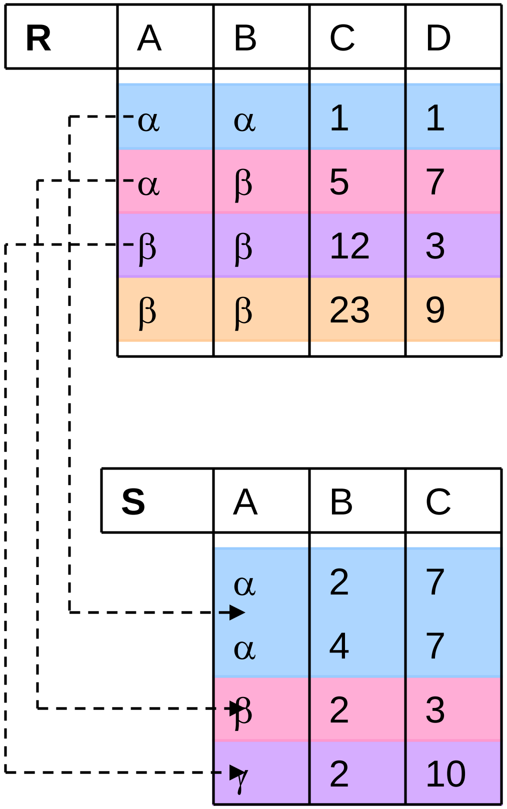
$$(\forall t)(\text{GIAOVIEN}(t) \wedge (t.\text{GVQLCM} \neq \text{null} \Rightarrow (\exists s)(\text{GIAOVIEN}(s) \wedge s.\text{MABM} = t.\text{MABM} \wedge s.\text{MAGV} = t.\text{GVQLCM})))$$

    - Influence table

IC_2	Insert	Delete	Update
GIAOVIEN	+	+	+ (GVQLCM, MABM)

IC – Multiple tuples through multiple relations

- IC is defined between tuples in multiple relations



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

S	A	B	C
	α	2	7
	α	4	7
	β	2	3
	γ	2	10

Example 13

- Each project must have at least one work
  - Context: DETAI, CONGVIEC
  - Content:

$$(\forall t)\ (DETAI(t) \land (\exists s)(CONGIVEC(s) \land t.MADT = s.MADT))$$

  - Influence table:

IC13	Insert	Delete	Update
DETAI	+	-	+(MADT)
CONGVIEC	-	+	+ (MADT)

Example 14

- Each department must have at least one lecturer

  - Context: GIAOVIEN, BOMON
  - Content:

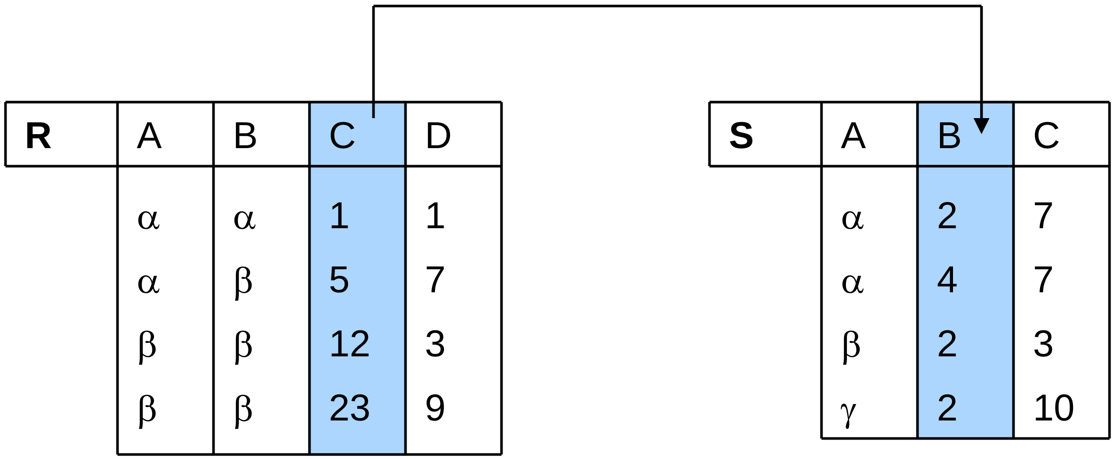
$$(\forall t)\ (\text{BOMON}(t) \wedge (\exists s)(\text{GIAOVIEN}(s) \wedge t.\text{MABM} = s.\text{MABM}))$$

  - Influence table:

IC14	Insert	Delete	Update
BOMON	+	-	+(MABM)
GIAOVIEN	-	+	+ (MABM)

IC – Multiple attributes through multiple relation

- IC is defined between attributes across multiple relations



R	A	B	C	D
	α	α	1	1
	α	β	5	7
	β	β	12	3
	β	β	23	9

S	A	B	C
	α	2	7
	α	4	7
	β	2	3
	γ	2	10

Example 15

- The birthday of the department head must be less than the date of acceptance of the head position

  - Context: GIAOVIEN, BOMON
  - Content:

$$(\forall t)(BOMON(t) \wedge (\forall s)(GIAOVIEN(s) \wedge (s.MAGV = t.TRUONGBM \Rightarrow s.NGAYSINH < t.NGAYNHANCHUC)))$$

  - Influence table:

IC$_{15}$	Insert	Delete	Update
GIAOVIEN	-	–	+ (NGAYSINH)
BOMON	+	–	+ (NGAYNHANCHUC, TRUONGBM)

Example 16

- Allowance of a work must be smaller than the budget of the project.

  - Bối cảnh: THAMGIADT, DETAI
  - Biểu diễn:

$$(\forall t)(THAMGIADT(t) \wedge (\forall s)(DETAI(s) \wedge (s.MADT = t.MADT \Rightarrow t.PHUCAP < s.KINHPHI)))$$

  - Bảng tầm ảnh hưởng:

R_{16}	Thêm	Xóa	Sửa
THAMGIADT	+	–	+ (PHUCAP)
DETAI	-	–	+ (KINHPHI)

IC - Inference/calculated attribute

- Calculated attribute (also called inference attribute)
  - Its value is calculated from other attributes
- When a database has a calculated attribute
  - IC ensures the link between the calculated attributes with the source attributes is consistently maintainted

Example 17

- **BOMON**(MABM, TENBM, TRUONGBM, NGAYNHANCHUC, SO_GV)
- The attribute SO_GV must be equal with the number of lecturers of the department
  - Context: GIAOVIEN, BOMON
  - Content:

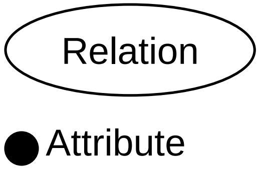
$$(\forall t)(BOMON(t) \wedge t.SO\_GV = \textbf{card}(\{\ s \mid GIAOVIEN(s) \wedge s.MABM = t.MABM\}))$$

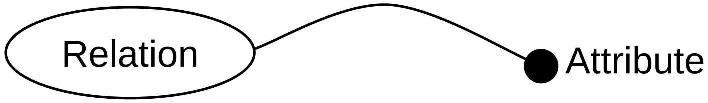
  - Influence table:

IC_{17}	Insert	Delete	Update
GIAOVIEN	+	+	+ (MABM)
BOMON	–	–	+ (SO_GV)

IC – Relations form a cycle in relation graph

- Database schema can be presented by a graph
  - Vertex (đỉnh)
    - Relation
    - Attribute
  - Edge
    - A line connecting a relation vertex to an attribute vertex in the database schema

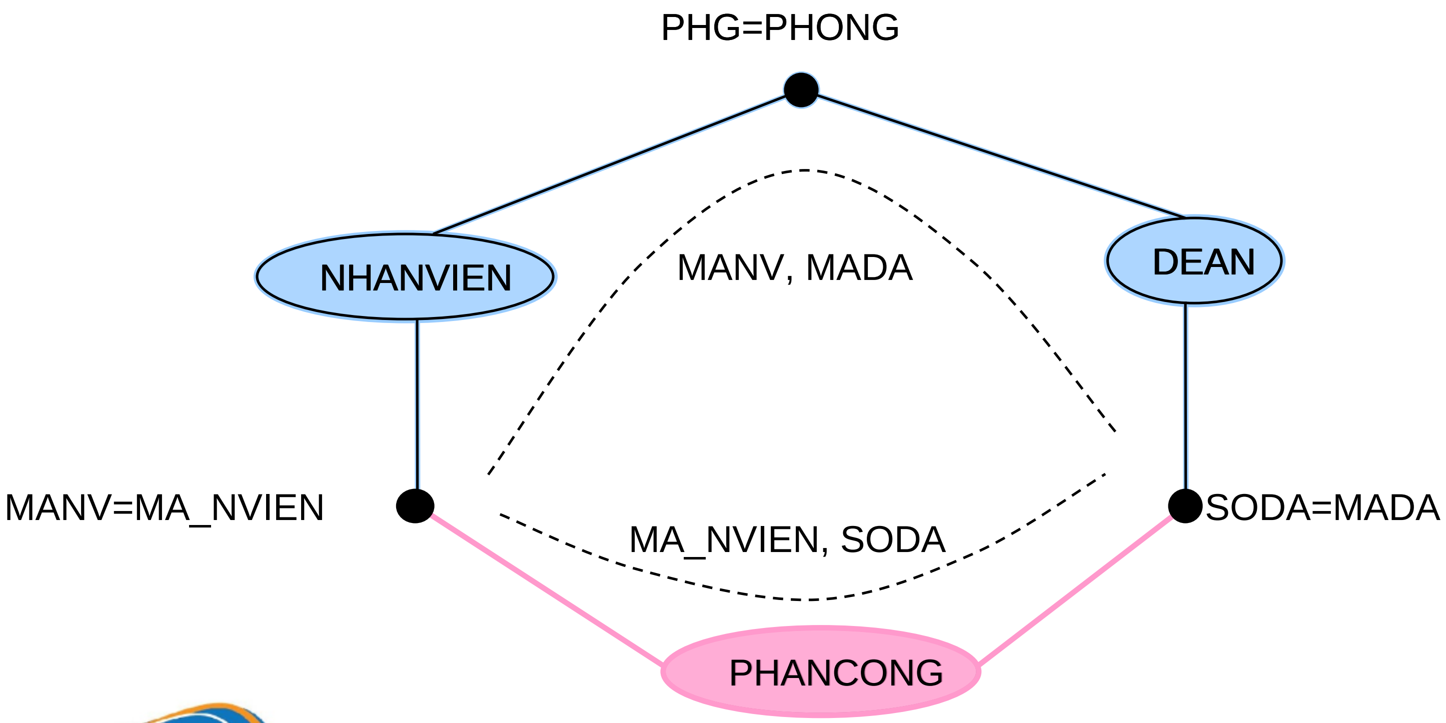




- Cycle
  - Graph appears closed path ~ Acyclic database schema

Example 17

- Employees are only assigned to projects in charged by their departments



Example 17 (cont.)

- Context: NHANVIEN, DEAN, PHANCONG
- Content:

$$NVDA \leftarrow NHANVIEN \bowtie_{PHG=PHONG} DEAN$$

$$(\forall t)\ (PHANCONG(t) \wedge (\exists s)(NVDA(s) \wedge$$
$$t.MA\_NVIEN = s.MANV \wedge t.MADA = s.SODA\ ))$$

- Influence table:

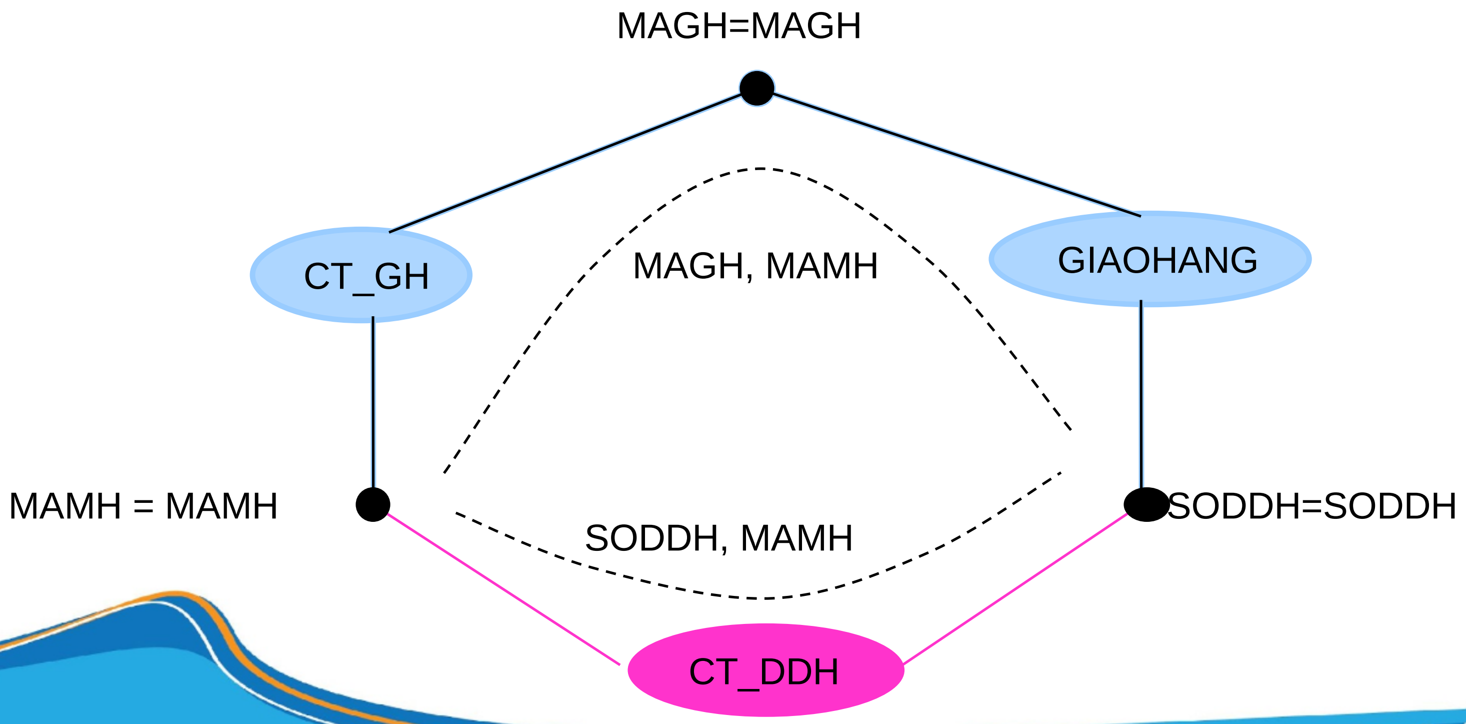
IC_{17}	Insert	Delete	Update
NHANVIEN	–	–	+ (MANV,PHG)
DEAN	–	–	+ (MADA,PHONG)
PHANCONG	+	–	+ (MA_NVIEN,SODA)

fit@hcmus

Example 18

- DDH (<u>SODDH</u>, NGAYDH, MAKH)
- CT_DDH (<u>SODDH, MAMH</u>, SOLUONG, DONGIA)
- GIAOHANG(<u>MAGH</u>, NGAYGH, TONGTIEN, SODDH)
- CT_GH (<u>MAGH, MAMH</u>)

IC: Only products ordered by customer are allowed to be delivered.



Content

- Overview
- Characteristics of Integrity Constraints
- Classification
- **Implementation**
  - Assertion
  - Trigger
  - Transaction (giao tác)
  - Stored Procedure (thủ tục lưu trữ nội)
  - Application (ứng dụng)

Implementation

- The ICs are implemented by:
  - Primary key
  - Foreign key
  - Check constraint
  - Assertion
  - Trigger
  - Transaction

Assertion

- Is an SQL expression that always returns the value TRUE at all times.
  - Users need to tell what must be true

- Syntax

```
CREATE ASSERTION <Tên_assertion> CHECK (<Điều_kiện>)
```

```
DROP ASSERTION <Tên_assertion>
```

Example 15

- Ngày sinh của trưởng bộ môn phải nhỏ hơn ngày nhận chức

```
CREATE ASSERTION R12 CHECK (
        NOT EXISTS (
                SELECT *
                FROM GIAOVIEN, BOMON
                WHERE MAGV=TRUONGBM
                AND NGAYSINH > NGAYNHANCHUC )
)
```

Example 19

- Lương của trưởng bộ môn phải lớn hơn 50000

```
CREATE ASSERTION R15 CHECK (
        NOT EXISTS (
                SELECT *
                FROM GIAOVIEN, BOMON
                WHERE MAGV=TRUONGBM
                AND LUONG < 50000 )
)
```

Example 19 (cont.)

- Lương của trưởng bộ môn phải lớn hơn 50000

Check Constraint

```
ALTER TABLE BOMON (
            TENBM NVARCHAR(50) UNIQUE,
            MABM CHAR(10) NOT NULL,
            TRUONGBM CHAR(10),
            NGAYNHANCHUC DATETIME,
            CONSTRAINT CHK_BM_LUONGTRUONGBM CHECK (
                        TRUONGBM NOT IN (SELECT MAGV FROM
GIAOVIEN
                                                WHERE LUONG <= 50000 ))
)
```

Example 16

- Số lượng giáo viên của mỗi bộ môn không quá 20 người

```
CREATE ASSERTION R16 CHECK (
          20 >= ALL ( SELECT COUNT(MAGV)
                      FROM GIAOVIEN
                      GROUP BY MABM )
)
```

Example 16 (cont.)

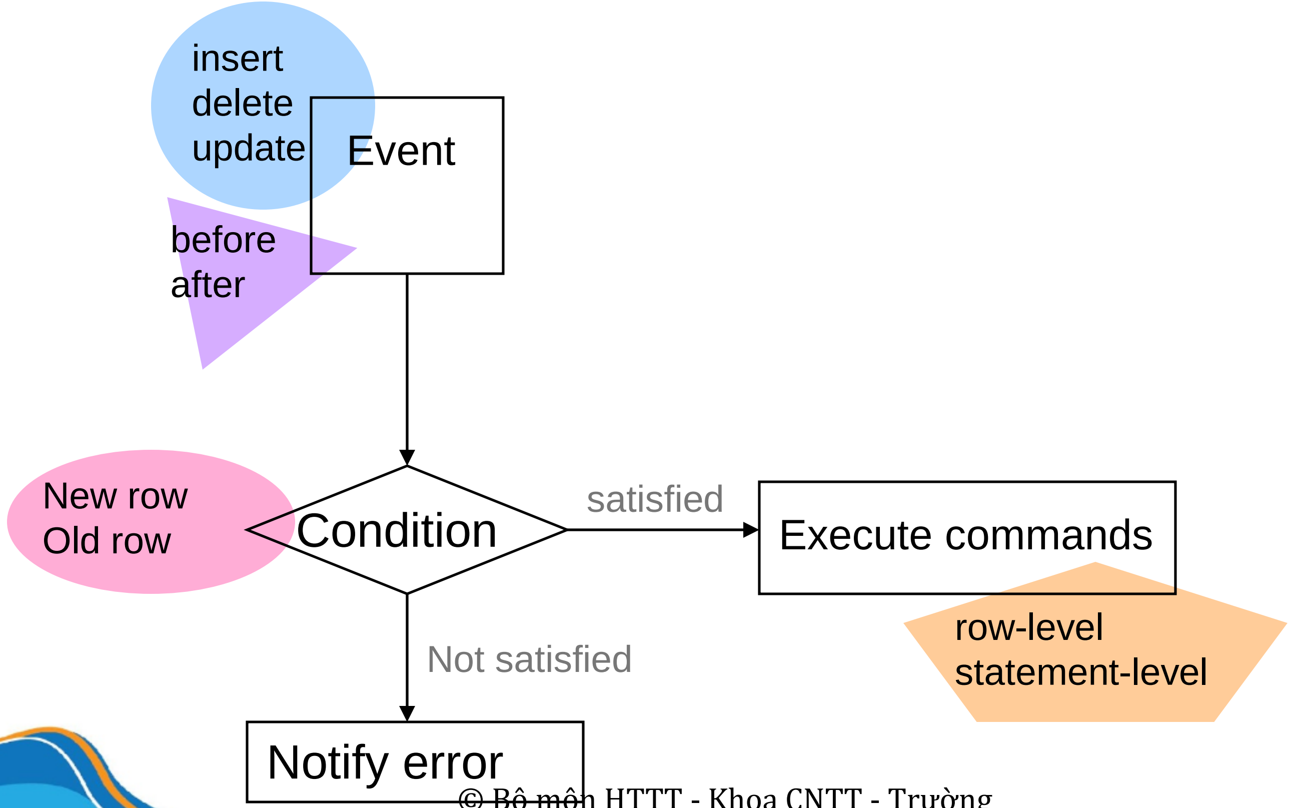
- Số lượng giáo viên của mỗi bộ môn không quá 20 người

Check Constraint

```
ALTER TABLE GIAOVIEN ADD
CONSTRAINT CHK_GV_SLGVBM CHECK (
          20 >= ALL ( SELECT COUNT(MAGV) FROM GIAOVIEN
                         GROUP BY MABM ))
```

Trigger

- A set of commands that are automatically executed when an event occurs in the database



Trigger (cont.)

- Syntax

```
CREATE TRIGGER <Trigger_name>
AFTER|BEFORE    INSERT|UPDATE|DELETE    ON   <Table>
REFERENCING
            NEW   ROW|TABLE   AS   <Name_1>
            OLD   ROW|TABLE   AS   <Name_2>
FOR EACH ROW | FOR EACH STATEMENT
WHEN (<Condition>)
            <Tập_lệnh_SQL>
```

```
DROP TRIGGER <Trigger_name>
```

Example 15

- The salary of the department head must be greater than 50000

```
CREATE TRIGGER TR_BM_UPD
AFTER UPDATE OF TRUONGBM ON BOMON
REFERENCING
          NEW ROW AS NewTuple
FOR EACH ROW
WHEN (50000 >= (SELECT LUONG          FROM GIAOVIEN
                              WHERE MAGV=NewTuple.TRUONGBM))
          Notify error to users
```

Example 15 (cont.)

- The salary of the department head must be greater than 50000

```
CREATE TRIGGER TR_BM_UPD
AFTER UPDATE OF TRUONGBM ON BOMON
REFERENCING
       NEW ROW AS NewTuple
       OLD ROW AS OldTuple
FOR EACH ROW
WHEN (50000 >= (SELECT LUONG      FROM GIAOVIEN
                    WHERE MAGV=NewTuple.TRUONGBM))
       UPDATE BOMON
       SET TRUONGBM=OldTuple.TRUONGBM
       WHERE TRUONGBM=NewTuple.TRUONGBM
```

Example 15 (cont.)

- The salary of the department head must be greater than 50000

```
CREATE TRIGGER TR_BM_UPD
AFTER UPDATE OF LUONG ON GIAOVIEN
REFERENCING
          NEW ROW AS NewTuple
          OLD ROW AS OldTuple
FOR EACH ROW
WHEN (NewTuple.LUONG <= 50000 AND NewTuple.MAGV IN (
                    SELECT TRUONGBM FROM BOMON ))
          UPDATE GIAOVIEN
          SET LUONG=OldTuple.LUONG
          WHERE LUONG=NewTuple.LUONG
```

Transaction

- A set of instructions that perform a task/transaction in a database, such that
  - Or all instructions are executed successfully
  - Or no instruction is executed

- Ex: transfer money in a bank

```
Begin transaction  Chuyển_tiền
          Giảm tiền trong tài khoản người gửi
          Tăng tiền trong tài khoản người nhận
          Nếu tất cả đều thành công thì complete
          Ngược lại roll back
End transaction
```

Transaction (cont.)

- A transaction ensure
  - Atomicity (Tính nguyên tố)
  - Consistency (Tính nhất quán của CSDL)
    - Related ICs are not violated
      - While executing the transaction
      - Before and after executing the transaction

Example

- Each match is a competition of exactly 2 teams

```
Giao tác Thêm_trận_đấu(t, s)
         Thêm t vào THIDAU
         Thêm s vào THIDAU
         Nếu có một thao tác thất bại thì
                 Quay lui giao tác
         Ngược lại
                 Hoàn tất giao tác
         Cuối nếu
Cuối giao tác
```

Example

```
Giao tác Xóa_trận_đấu(ngay, gio)
    Với mọi s$\in$THIDAU (s.NGAY=ngay $\wedge$ s.GIO=gio)
        Xóa s khỏi THIDAU
    Cuối với mọi
    Nếu có một thao tác thất bại thì
        Quay lui giao tác
    Ngược lại
        Hoàn tất giao tác
    Cuối nếu
Cuối giao tác
```

Example

- Each invoice must have at least one invoice detailed line

```
Giao tác Thêm_hóa_đơn
          Thêm HOADON
          Thêm chi tiết thứ 1 vào CTHD
          Thêm chi tiết thứ 2 vào CTHD
          …
          Nếu có một thao tác thêm thất bại thì
                    Quay lui giao tác
          Ngược lại
                    Hoàn tất giao tác
          Cuối nếu
Cuối giao tác
```

Example

```
Giao tác Thêm_hóa_đơn
         Thêm HOADON
         Thêm chi tiết thứ 1 vào CTHD
         Thêm chi tiết thứ 2 vào CTHD
         …
         Nếu có một thao tác thêm thất bại thì
                  Quay lui giao tác
         Ngược lại
                  Hoàn tất giao tác
         Cuối nếu
Cuối giao tác
```

Stored Procedure

- DBMSs provide a way to store functions or procedures
  - Stored in the database schema
  - Used in SQL statements

- Syntax

```
CREATE PROCEDURE <Tên_thủ_tục> <DS_tham_số>
AS
            Khai báo biến cục bộ
            Thân chương trình
GO
EXEC <Tên_thủ_tục> <DS_ tham_số>
```

Example

- Each match is a competition of exactly 2 teams

```
CREATE PROCEDURE Thêm_trận_đấu
t THIDAU , s THIDAU
AS
        begin tran
                Thêm t vào THIDAU
                If @@error<>0 rollback tran

                Thêm s vào THIDAU
                If @@error<>0    rollback tran
        commit tran
GO

EXEC Thêm_trận_đấu  x, y
```

Notice

- DBMS will check ICs
  - After an update operation has taken place on the database
  - At the end of each transaction
- Where should an IC be implemented?
  - DBMS
  - Application

  - Too much Trigger → slow system
  - Stored Procedure → high performance