**Bài thực hành số 2**

Cho lược đồ cơ sở dữ liệu sau:

Khuvuc(**IP**, tenKhuvuc, tang )

Phòng (**MP**, tenphong, somay , *IP* )

May (**idMay**, tenmay, *IP*, ad, *idloai*, *MP* )

Phanmem(**idPM**, tenPM, ngaymua, version, *idloai*, gia )

Caidat (**id***, idMay*, *idPM*, ngaycai)

Loai (**idloai**, tenloai )

Chạy script đã thực hiện trong bài thực hành số 1 để tạo CSDL và thêm dữ liệu cho CSDL vừa tạo.

Viết script hoten\_bt2.sql bao gồm các câu trả lời cho các câu hỏi sau :

1. Loại của máy 'p8'

* **SQL:**

select idMay, tenloai

from Loai l, May m

where l.idloai=m.idloai and idMay = 'p8';

* **Đại số quan hệ:**

tenloai(Loai \* idMay = “p8” (May))

1. Tên của các phần mềm 'UNIX'

* **SQL:**

select tenPM, idloai

from Phanmem

where idloai = 'UNIX';

* **Đại số quan hệ:**

tenPM( idLoai = “UNIX” (Phanmem))

1. Tên phòng, địa chỉ IP phòng, mã phòng của các máy loại 'UNIX' hoặc 'PCWS'

* **SQL:**

select m.MP, tenphong, m.IP, m.idMay, idloai

from Phong p, May m

where p.MP = m.MP

and m.idloai = 'UNIX' or m.idloai = 'PCWS' ;

* **Đại số quan hệ:**

tenphong, IP, MP(Phong \* iloai = “UNIX” v idLoai = “PCWS” (May))

1. Tên phòng, địa chỉ IP phòng, mã phòng của các máy loại 'UNIX' hoặc 'PCWS' ở khu vực '130.120.80', sắp xếp kết quả tăng dần theo mã phòng

* **SQL:**

select m.MP, tenphong, m.IP, m.idMay, idloai

from Phong p, May m

where p.MP = m.MP

and m.idloai = 'UNIX' or m.idloai = 'PCWS' and m.IP = '130.120.80' order by p.MP ASC ;

* **Đại số quan hệ:**

tenphong, IP, MP (Phong \* iloai = “UNIX” v idLoai = “PCWS” ^ IP = “130.20.80” (May))

1. Số các phần mềm được cài đặt trên máy 'p6'

* **SQL:**

select count(idMay) as Tong\_PM\_caidat

from Caidat

where idMay = 'p6';

* **Đại số quan hệ:**

COUNTidPM (idMay = “p6” (Caidat))

1. Số các máy đã cài phần mềm 'log1'

* **SQL:**

select count(idPM) as Tong\_so\_may

from Caidat

where idPM = 'log1';

* **Đại số quan hệ:**

COUNTidPM ( idPM= “log1” (Caidat))

1. Tên và địa chỉ IP (ví dụ: 130.120.80.1) đầy đủ của các máy loại 'TX'

* **SQL:**

select idMay, tenloai, IP

from Loai l, May m

where l.idloai=m.idloai and m.idloai = 'TX';

* **Đại số quan hệ:**

tenloai, IP (Loai \* iloai = “TX” (May))

1. Tính số phần mềm đã cài đặt trên mỗi máy

* **SQL:**

select idMay, count(idPM) as so\_Phan\_men

from Caidat

group by idMay;

* **Đại số quan hệ:**

idMayG(COUNTidPM (Caidat)

1. Tính số máy mỗi phòng

* **SQL:**

select m.MP, count(m.idMay) as so\_may

from Phong p, May m

where p.MP=m.MP

group by m.MP;

* **Đại số quan hệ:**

MPG(COUNTidMay (MP(May \* Phong))

1. Tính số lần cài đặt của mỗi phần mềm trên các máy khác nhau

* **SQL:**

select p.idPM, tenPM, count(c.id) as so\_lan\_cai\_dat

from Phanmem p, Caidat c

where c.idPM=p.idPM

group by p.idPM, tenPM;

* **Đại số quan hệ:**

MPG(COUNTid (idPM, tenPM(Phanmem \* Caidat))

1. Giá trung bình của các phần mềm UNIX

* **SQL:**

select AVG(gia)

from Phanmem

where idloai = 'UNIX';

* **Đại số quan hệ:**

AVGgia (idloai = “UNIX” (Phanmem))

1. Ngày mua phần mềm gần nhất

* **SQL:**

select max(ngaymua) as Ngay\_Mua\_GanNhat

from Phanmem;

* **Đại số quan hệ:**

MAxgia (Phanmem)

1. Số máy có ít nhất 2 phần mềm

* **SQL:**

select p.idPM, tenPM, count(c.id) as so\_lan\_cai\_dat

from Phanmem p, Caidat c

where c.idPM=p.idPM

group by p.idPM, tenPM

HAVING COUNT(c.id) >= 2;

* **Đại số quan hệ:**

idPM, tenPMG(COUNTid >=2 (idPM, tenPM(Phanmem \* Caidat))

1. Số máy có ít nhất 2 phần mềm, sử dụng một select con trong mệnh đề FROM

* **SQL:**

select \*

from (select idmay, count(idmay) as So\_PhanMem

from caidat

group by idmay)

where So\_PhanMem >= 2;

* **Đại số quan hệ:**

**\*\*Các câu 15) và 17) viết bằng 2 cách : SELECT con và OUTER JOIN**

1. Tìm các loại không thuộc loại máy

* **SQL:**

select idloai

from Loai

where idloai not in (select idloai from May);

* **Đại số quan hệ:**

1. Tìm các loại thuộc cả hai loại máy và loại phần mềm

* **SQL:**

select idloai

from Loai

where idloai in (select idloai from May);

* **Đại số quan hệ:**

1. Tìm các loại máy không phải là loại phần mềm

* **SQL:**

select DISTINCT idloai

from May

where idloai not in (select idloai from Phanmem);

* **Đại số quan hệ:**

**\*\*Viết các câu từ 18) đến 22) sử dụng SELECT lồng nhau**

1. Địa chỉ IP đầy đủ của các máy cài phần mềm 'log6'

* **SQL:**

select idMay, IP

from May

where idMay in (select idMay from Caidat where idPM = 'log6');

* **Đại số quan hệ:**

1. Địa chỉ IP đầy đủ của các máy cài phần mềm tên 'Oracle 8'

* **SQL:**

select idMay, IP

from May

where idMay in (select c.idMay

from Caidat c, Phanmem p

where c.idPM = p.idPM

and tenPM = 'Oracle 8');

* **Đại số quan hệ:**

1. Tên của các khu vực có chính xác 3 máy loại 'TX'

* **SQL:**

select tenKhuvuc

from Khuvuc

where IP in (select IP

from May

where idloai= 'TX'

group by IP

having count(IP)=3 );

* **Đại số quan hệ:**

1. Tên phòng có ít nhất một máy cài phần mềm tên 'Oracle 6'

* **SQL:**

select tenphong

from Phong ph, May m

where ph.MP = m.MP

and m.idMay in (select idMay

from Caidat c, Phanmem p

where c.idPM = p.idPM

and tenPM= 'Oracle 6')

group by tenphong

having count(tenphong) >= 1;

* **Đại số quan hệ:**

1. Tên phần mềm được mua gần nhất (sử dụng câu 12)

* **SQL:**

select tenPM, ngaymua

from Phanmem

where ngaymua = (select max(ngaymua) from Phanmem);

* **Đại số quan hệ:**

**\*\*Viết các câu từ 18) đến 21) sử dụng JOIN, đánh số cho các câu hỏi này từ 24) đến 26)**

1. Tên của máy có ít nhất một phần mềm chung với máy 'p6' → kq: p2, p8, p1

* **SQL:**

select c.idMay, tenmay

from Caidat c

Join May m

on c.idMay = m.idMay

where idPM in (select idPM from Caidat where idMay = 'p6')

and c.idMay <> 'p6';

* **Đại số quan hệ:**

1. Tên của phần mềm PCNT có giá lớn hơn bất kỳ giá của một phần mềm UNIX nà

* **SQL:**

SELECT p.tenPM

FROM Phanmem p

WHERE p.idloai = 'PCNT'

AND p.gia > ANY (SELECT gia FROM Phanmem WHERE idloai = 'UNIX');

* **Đại số quan hệ:**

1. Tên của phần mềm UNIX có giá lớn hơn tất cả các giá của các phần mềm PCNT

* **SQL:**

SELECT p.tenPM

FROM Phanmem p

WHERE p.idloai = 'UNIX'

AND p.gia > ALL (SELECT gia FROM Phanmem WHERE idloai = 'PCNT');

* **Đại số quan hệ:**

1. Tên của các máy có cùng phần mềm như máy 'p6' (có thể nhiều phần mềm hơn máy 'p6') → kq: p2, p8

* **SQL:**

select c.idMay, tenmay

from Caidat c

Join May m

on c.idMay = m.idMay

where idPM in (select idPM from Caidat where idMay = 'p6')

and c.idMay <> 'p6';

* **Đại số quan hệ:**

1. Tên của các máy có chính xác các phần mềm như máy 'p2' → kq: p8

* **SQL:**

select c.idMay, tenmay

from Caidat c

Join May m

on c.idMay = m.idMay

where idPM in (select idPM from Caidat where idMay = 'p2')

and c.idMay <> 'p2';

* **Đại số quan hệ:**

**Bài thực hành số 3**

Cho lược đồ cơ sở dữ liệu về thư viện như sau:

TACPHAM(**NT**, tua, tacgia)

DOCGIA (**ND**, ho, ten, dchi, tel)

SACH (**NS**, nxb, *NT*)

MUON (**NS**, **ngaymuon**, hantra, ngaytra, *ND*);

Câu 1: Chạy script CSDLBib.sql để tạo CSDL trên và thêm dữ liệu cho các bảng vừa tạo

Câu 2: Viết script hoten\_bt3.sql bao gồm các câu trả lời cho các câu hỏi sau :

1. Các tác phẩm (NT, tựa) của tác giả 'Guy de Maupassant'.

* **SQL:**

select NT, tua

from TACPHAM

where tacgia = 'Guy de Maupassant';

* **Đại số quan hệ:**

NT, tua( tacgia = “Guy de Maupassant” (TACPHAM))

1. Các độc giả sống ở địa chỉ '32 rue des Alouettes, 75003 Paris'.

* **SQL:**

select \*

from DOCGIA

where dchi = '32 rue des Alouettes, 75003 Paris';

* **Đại số quan hệ:**

dchi = “32 rue des Alouettes, 75003 Paris” (DOCGIA)

1. Tìm tên nhà xuất bản các tác phẩm bao gồm từ 'Fleur'

* **SQL:**

select nxb, tua

from TACPHAM tp

LEFT JOIN SACH s

ON tp.NT = s.NT

where tua LIKE '%Fleur%';

* **Đại số quan hệ:**

NT, tua(( tua = “%Fleur%” (TACPHAM))\* SACH)

1. Tìm tên các tác phẩm bắt đầu bằng 'Le'

* **SQL:**

select tua

from TACPHAM

where tua LIKE 'Le%';

* **Đại số quan hệ:**

tua( tua = “Le%” (TACPHAM))

1. Tìm tên các độc giả có mượn sách trong trong khoảng thời gian từ ngày 15/9/2007 đến 20/09/2007

* **SQL:**

select ten, ngaymuon

from MUON m

LEFT JOIN DOCGIA dg ON m.ND=dg.ND

where ngaymuon between TO\_DATE('15/9/2007', 'DD-MM-YYYY') and TO\_DATE('20/09/2007', 'DD-MM-YYYY');

* **Đại số quan hệ:**

ten, ngaymuon(( ngaymuon >= “15/09/2007” ^ ngaymuon <= “20/09/2007” (MUON))\* DOCGIA)

1. Số tác phẩm có trong thư viện.

* **SQL:**

select count(\*) as so\_tacpham

from TACPHAM;

* **Đại số quan hệ:**

COUNTNT(TACPHAM)

1. Tính Số tác phẩm của mỗi tác giả.

* **SQL:**

select tacgia, count(\*) as so\_tacpham

from TACPHAM

group by tacgia;

* **Đại số quan hệ:**

tacgiaG(COUNTNT(tacgia(TACPHAM)))

1. Tính Số tác giả có ít nhất hai tác phẩm

* **SQL:**

select tacgia, count(\*) as so\_tacpham

from TACPHAM

group by tacgia

having count(\*) >=2;

* **Đại số quan hệ:**

tacgiaG(COUNTNT >= 2 (tacgia(TACPHAM)))

1. Tính Số sách của mỗi tác phẩm.

* **SQL:**

select tua, count(\*)as SL\_Sach

from TACPHAM tp

JOIN SACH s ON tp.NT = s.NT

group by tua;

* **Đại số quan hệ:**

tuaG(COUNTNT (tua(TACPHAM) \* SACH))

1. Tìm tên tác phẩm có nhiều sách nhất

* **SQL:**

select tua, count(\*) as SL\_Sach

from TACPHAM tp

JOIN SACH s ON tp.NT = s.NT

group by tua

having count(\*) = (select Max(count(\*)) as SL\_Sach from SACH group by NT);

* **Đại số quan hệ:**

R = tuaG(COUNTNS:sosach (TACPHAM \* SACH))

sosach = Max(R) (R)

1. Tên nhà xuất bản xuất bản nhiều sách nhất

* **SQL:**

select nxb, count(\*) as SL\_Sach

from SACH

group by nxb

having count(\*) = (select Max(count(\*)) as SL\_Sach from SACH group by nxb);

* **Đại số quan hệ:**

R = nxbG(COUNTNS:sosach\_nxb (SACH))

sosach\_nxb = Max(R) (R)

1. Nhà xuất bản của tác phẩm tiêu đề 'Germinal'.

* **SQL:**

select DISTINCT nxb, tua

from TACPHAM tp

JOIN SACH s ON tp.NT = s.NT

where tua = 'Germinal '

* **Đại số quan hệ:**

nxb, tua(tua = “Germinal”(TACPHAM) \* SACH )

1. Tựa của tác phẩm mà có ít nhất hai quyển sách.

* **SQL:**

select tua, count(\*) as SL\_Sach

from TACPHAM tp

JOIN SACH s ON tp.NT = s.NT

group by tua

having count(\*) > = 2;

* **Đại số quan hệ:**

tuaG(COUNTNT >= 2 (TACPHAM \* SACH)

1. Tên độc giả đã mượn tác phẩm 'Poésie'.

* **SQL:**

select ho, ten, tua

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where tua = 'Poésie';

* **Đại số quan hệ:**

ho,ten(DOCGIA \* MUON \* SACH \* (tua = “Poésie”(TACPHAM)))

1. Những độc giả nào đã mượn tác phẩm Les 'Fleurs du mal'

* **SQL:**

select ho, ten, tua

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where tua = 'Les Fleurs du mal';

* **Đại số quan hệ:**

ho,ten(DOCGIA \* MUON \* SACH \* (tua = “'Les Fleurs du mal”(TACPHAM)))

1. Tìm số lần mượn sách của mỗi độc giả theo năm

* **SQL:**

select m.ND, ten, EXTRACT(YEAR FROM ngaymuon) as Nam, count(m.ND) as Solan\_Muon

from MUON m

LEFT JOIN DOCGIA dg ON m.ND = dg.ND

group by m.ND, EXTRACT(YEAR FROM ngaymuon), ten;

* **Đại số quan hệ:**

ND, ngaymuon, tenG(COUNTND(ND,ten,ngaymuon(MUON \* DOCGIA))

1. Tìm tựa sách, tên độc giả của các độc giả chưa trả sách

* **SQL:**

select tua, ten, ngaymuon, ngaytra

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ngaytra is null

* **Đại số quan hệ:**

tua, ten(DOCGIA \* (ngaytra = “”(MUON)) \* SACH \* TACPHAM)

1. Tìm các tựa sách, tên độc giả của các độc giả trả sách quá thời hạn cho phép

* **SQL:**

select tua, ten, ngaymuon, hantra, ngaytra

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ngaytra not in (select ngaytra

from muon

where ngaytra between ngaymuon and hantra)

* **Đại số quan hệ:**

R = ngaytra(ngaytra >= ngaymuon ^ ngaytra <= hantra(MUON))

tua, ten(DOCGIA \* (ngaytra = “”(MUON)) \* SACH \* TACPHAM)

1. Tìm các tựa sách, tên độc giả của các độc giả trả sách trước thời hạn

* **SQL:**

select tua, ten, ngaymuon, hantra, ngaytra

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ngaytra < hantra

* **Đại số quan hệ:**

tua,ten(DOCGIA \* (ngaytra < hantra(MUON))\* SACH \* TACPHAM)

1. Tìm tựa tác phẩm có nhiều người mượn nhất

* **SQL:**

select tua, count(\*)as SL\_Muon

from MUON m

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

group by tua

having count(\*) = (select max(count(\*))

from MUON m

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

group by tua)

* **Đại số quan hệ:**

1. Tìm tựa tác phẩm có ít người mượn nhất

* **SQL:**

select tua, count(\*)as SL\_Muon

from MUON m

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

group by tua

having count(\*) = (select min(count(\*))

from MUON m

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

group by tua)

* **Đại số quan hệ:**

1. Tìm độc giả mượn nhiều tác phẩm nhất

* **SQL:**

select ten, count(\*)as SL\_TacPHam

from DOCGIA dg

JOIN MUON m ON dg.ND = m.ND

group by ten

having count(\*) = (select max(count(\*))

from DOCGIA dg

JOIN MUON m ON dg.ND = m.ND

group by m.ND)

* **Đại số quan hệ:**

1. Tìm tên tên độc giả mượn ít tác phẩm nhất

* **SQL:**

select ten, count(\*)as SL\_TacPHam

from DOCGIA dg

JOIN MUON m ON dg.ND = m.ND

group by ten

having count(\*) = (select min(count(\*))

from DOCGIA dg

JOIN MUON m ON dg.ND = m.ND

group by m.ND)

* **Đại số quan hệ:**

1. Tên độc giả đã mượn tác phẩm của 'Victor Hugo'

* **SQL:**

select ho, ten, tua, tacgia

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where tacgia = 'Victor Hugo';

* **Đại số quan hệ:**

ho,ten(DOCGIA \* MUON \* SACH \* (tacgia = “Victor Hugo” (TACPHAM)))

1. Tên độc giả và các tác phẩm đã được mượn năm 2007

* **SQL:**

select ten, tua, ngaymuon

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where EXTRACT(YEAR FROM ngaymuon) = '2007';

* **Đại số quan hệ:**

ten, tua(DOCGIA \* (YEAR(ngaymuon) = “2007” (MUON) \* SACH \* TACPHAM))

1. Tìm tên tác phẩm có ít nhất một quyển sách không ai mượn

* **SQL:**

select tua, count(\*)

from MUON m

RIGHT JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ngaymuon is null

group by tua

having count(\*) >= 1;

* **Đại số quan hệ:**

tuaG(COUNTNT >= 1 (MUON \* SACH \* TACPHAM)

1. Tìm tên tác phẩm có tất cả các quyển sách đều được mượn

* **SQL:**

SELECT TP.tua AS TenTacPham

FROM TACPHAM TP

JOIN SACH S ON TP.NT = S.NT

LEFT JOIN MUON M ON S.NS = M.NS

GROUP BY TP.tua

HAVING COUNT(S.NS) > 0

AND COUNT(M.NS) = COUNT(S.NS);

* **Đại số quan hệ:**

tuaG(COUNTNS >0 (TACPHAM \* SACH \* MUON)

1. Tìm họ tên độc giả chưa mượn quyển sách nào

* **SQL:**

select ho, ten

from DOCGIA dg

LEFT JOIN MUON m ON dg.ND = m.nd

where ngaymuon is null;

* **Đại số quan hệ:**

ho,ten( ngaymuon = “” (MUON) \* DOCGIA)

1. Tìm thông tin về nhà suất bản của quyển sách mà chưa được ai mượn

* **SQL:**

select s.NS, nxb, tua, tacgia

from MUON m

RIGHT JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ngaymuon is null;

* **Đại số quan hệ:**

nxb, tua((ngaymuon = “” (MUON)) \* SACH \* TACPHAM)

1. Tìm tên tác phẩm có ít nhất 3 sách

* **SQL:**

select tua, count(\*) as SL\_Sach

from TACPHAM tp

JOIN SACH s ON tp.NT = s.NT

group by tua

having count(\*) > = 3;

* **Đại số quan hệ:**

tuaG(COUNTNT >= 3 (TACPHAM \* SACH)

1. Tìm thông tin về độc giả có mượn tất cả các quyển sách của tác phẩm 'Madame Bovary'

* **SQL:**

select ho, ten, tua, tacgia

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where tua = 'Madame Bovary';

* **Đại số quan hệ:**

ho,tem(DOCGIA \* MUON \* SACH \* (tua = “Madame Bovary” (TACPHAM)))

1. Tìm thông tin về độc giả có mượn các quyển sách giống như độc giả 'Raymond Carole'

* **SQL:**

select ho, ten, dchi, tel

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where tua in (select tua

from DOCGIA dg

JOIN MUON m ON dg.ND = m.nd

JOIN SACH s ON m.NS = s.NS

JOIN TACPHAM tp ON tp.NT = s.NT

where ten = 'Carole' and ho ='Raymond');

* **Đại số quan hệ:**

**Bài thực hành số 4**

Cho lược đồ cơ sở dữ liệu về phim như sau:

CANHAN (idCN , ho, ten)

FILM (idFilm, tua, nguoithuchien, kinhphi)

DIENVIEN (idFilm, idCN, thulao)

LOAI (idLoai, tenloai) ;

LOAIPHIM (idFilm, idLoai)

Câu 1. Chạy script CSDLFilm.sql để tạo CSDL và thêm dữ liệu cho CSDL vừa tạo.

Câu 2. Viết script hoten\_bt4.sql bao gồm các câu trả lời cho các câu hỏi sau :

1. Danh sách tất cả các bộ phim có khinh phí lớn hơn 1 triệu

* **SQL:**

select \*

from film

where kinhphi > 1000000;

* **Đại số quan hệ:**

kinhphi > “1000000” (Film)

1. Tên Loại của bộ phim tựa 'IMPORTANCE OF BEING ERNEST'

* **SQL:**

select tua, tenloai

from loai l, loaiphim lp, film f

where l.idLoai=lp.idLoai and f.idFilm=lp.idFilm

and tua = 'IMPORTANCE OF BEING ERNEST';

* **Đại số quan hệ:**

tenloai( tua = “IMPORTANCE OF BEING ERNEST” (Loai \* Loaiphim \* Film))

1. Tựa và kinh phí của các bộ phim thuộc thể loại 'SCIENCE FICTION' và có kinh phí lớn hơn 800,000 và nhỏ hơn 1 triệu. Sắp kết quả tăng dần theo kinh phí.

* **SQL:**

select tua, kinhphi, tenloai

from loai l, loaiphim lp, film f

where l.idLoai=lp.idLoai and f.idFilm=lp.idFilm

and tenloai = 'SCIENCE FICTION'

and kinhphi between 800000 and 1000000

order by kinhphi ASC;

* **Đại số quan hệ:**

tua,kinhphi(Loai \* Loaiphim \* (kinhphi >= “800000” ^ kinhphi <= “1000000” (Film))

1. Tên đầy đủ (ghép họ và tên), thù lao của các diễn viên đã tham gia bộ phim 'MARK OF THE DEVIL'. Sắp kết quả giảm dần theo thù lao

* **SQL:**

select ho|| ' ' ||ten as Hoten, thulao, tua

from film f

JOIN dienvien dv ON dv.idFilm = f.idFilm

JOIN canhan cn ON cn.idCN = dv.idCN

where tua = 'MARK OF THE DEVIL'

order by thulao DESC

* **Đại số quan hệ:**

1. Tên đầy đủ (ghép họ và tên) của người đã thực hiện bộ phim 'ONCE A THIEF'

* **SQL:**

select ho|| ' ' ||ten as Hoten, tua

from film f

JOIN dienvien dv ON dv.idFilm = f.idFilm

JOIN canhan cn ON cn.idCN = dv.idCN

where tua = 'ONCE A THIEF';

* **Đại số quan hệ:**

1. Tên các bộ phim bao gồm từ 'IMPOSSIBLE'

* **SQL:**

select tua

from film

where tua LIKE '%IMPOSSIBLE%';

* **Đại số quan hệ:**

tua( tua = “%IMPOSSIBLE%” (Film))

1. Thông tin về các bộ phim do viễn viên 'WILDE CORNEL' đóng. Sắp kết quả tăng dần theo họ và tên người thực hiện.

* **SQL:**

select tua, ho || ' ' || ten as DienVien, kinhphi, thulao

from film f

JOIN dienvien dv ON dv.idFilm = f.idFilm

JOIN canhan cn ON cn.idCN = dv.idCN

where ho = 'WILDE' and ten = 'CORNEL'

order by nguoithuchien ASC;

* **Đại số quan hệ:**

1. Tựa các bộ phim do 'WILSON HUGH'

* **SQL:**

select tua, ho || ' ' || ten as NguoiThucHien

from film f

JOIN canhan cn ON cn.idCN = f.nguoithuchien

where ho = 'WILSON' and ten = 'HUGH';

* **Đại số quan hệ:**

1. Họ tên cá nhân đã tham gia ít nhất 1 bộ phim do họ thực hiện

* **SQL:**

select ho,ten, count(\*) as so\_phim

from dienvien d

JOIN film f ON d.idFilm = f.idFilm

JOIN canhan c ON c.idCN = d.idCN

where d.idCN=f.nguoithuchien

group by ho,ten

having count(\*) >= 1

* **Đại số quan hệ:**

1. Họ tên cá nhân đã tham gia ít nhất 10 bộ phim do họ thực hiện

* **SQL:**

select ho,ten, count(\*) as so\_phim

from dienvien d

JOIN film f ON d.idFilm = f.idFilm

JOIN canhan c ON c.idCN = d.idCN

where d.idCN=f.nguoithuchien

group by ho,ten

having count(\*) >= 10

* **Đại số quan hệ:**

1. Tổng thù lao đã trả cho bộ phim 'TALE OF TWO CITIES'

* **SQL:**

select sum(thulao) as Tong\_Thulao

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

where tua = 'TALE OF TWO CITIES';

* **Đại số quan hệ:**

1. Số phim mỗi thể loại.

* **SQL:**

select tenloai, count(\*) as So\_Phim

from film f

JOIN loaiphim lp ON f.idFilm = lp.idFilm

JOIN loai l ON l.idLoai = lp.idLoai

group by tenloai;

* **Đại số quan hệ:**

1. Tên loại có nhiều phim nhất

* **SQL:**

select tenloai, count(\*) as So\_Phim

from film f

JOIN loaiphim lp ON f.idFilm = lp.idFilm

JOIN loai l ON l.idLoai = lp.idLoai

group by tenloai

having count(\*) = (select Max(count(\*)) as So\_Phim

from film f

JOIN loaiphim lp ON f.idFilm = lp.idFilm

group by idloai);

* **Đại số quan hệ:**

1. Kinh phí trung bình của phim theo người thực hiện

* **SQL:**

select ho, ten, ROUND(AVG(kinhphi), 2) as KinhPhi\_TrungBinh

from film f

JOIN canhan c ON f.nguoithuchien = c.idCN

group by ho, ten

order by KinhPhi\_TrungBinh DESC;

* **Đại số quan hệ:**

1. Họ tên người thực hiện đã thực hiện hơn 50 bộ phim

* **SQL:**

select ho || ' ' || ten as Hoten, count(\*) as SL\_Phim

from film f

JOIN canhan c ON f.nguoithuchien = c.idCN

group by ho,ten

having count(\*) > 50;

* **Đại số quan hệ:**

1. Họ tên người thực hiện thực hiện nhiều phim nhất

* **SQL:**

select ho || ' ' || ten as Hoten, count(\*) as SL\_Phim

from film f

JOIN canhan c ON f.nguoithuchien = c.idCN

group by ho,ten

having count(\*) = (select Max(count(\*)) as SL\_Phim

from film f

group by nguoithuchien);

* **Đại số quan hệ:**

1. Họ tên diễn viên đã tham gia hơn 50 bộ phim

* **SQL:**

select ho || ' ' || ten as Hoten, count(\*) as SL\_Phim

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN

group by ho,ten

having count(\*) > 50;

* **Đại số quan hệ:**

1. Họ tên diễn viên tham gia nhiều phim nhất

* **SQL:**

select ho || ' ' || ten as Hoten, count(\*) as SL\_Phim

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN

group by ho,ten

having count(\*) = (select Max(count(\*)) as SL\_Phim

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

group by idCN);

* **Đại số quan hệ:**

1. Tên bộ phim có kinh phí thực hiện cao nhất

* **SQL:**

select tua, kinhphi as Kinhphi\_caonhat

from film

where kinhphi = (select max(kinhphi) from film);

* **Đại số quan hệ:**

1. Tên bộ phim có kinh phí thực hiện thấp nhất

* **SQL:**

select tua, kinhphi as Kinhphi\_Thapnhat

from film

where kinhphi = (select min(kinhphi) from film);

* **Đại số quan hệ:**

1. Tên bộ phim đã trả thu lao cao nhất cho một diễn viên nào đó

* **SQL:**

select tua, thulao, ho || ' ' || ten as Dienvien

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN

where thulao = (select max(thulao)

from film f

JOIN dienvien d ON f.idFilm = d.idFilm);

* **Đại số quan hệ:**

1. Tên phim, kinh phí thực hiện và tổng thù lao đã trả cho các diễn viên của mỗi phim

* **SQL:**

select tua, kinhphi, thulao, ho || ' ' || ten as Dienvien

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN;

* **Đại số quan hệ:**

1. Họ tên, số lần tham gia và tổng thù lao đã nhận của mỗi diên viên có tổng thù lao lớn hơn 800.000

* **SQL:**

select ho || ' ' || ten as Hoten, count(\*) as SL\_Thamgia, sum(thulao) as Tong\_Thulao

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN

group by ho, ten

having sum(thulao) > 800000;

* **Đại số quan hệ:**

1. Tìm tên thể loại và trung bình khinh phí của mỗi thể loại mà có trung bình kinh phí lớn hơn 500 triệu

* **SQL:**

select tenloai, AVG(kinhphi) as TB\_KinhPhi

from film f

JOIN loaiphim lp ON f.idFilm = lp.idFilm

JOIN loai l ON l.idLoai = lp.idLoai

group by tenloai

having AVG(kinhphi) > 500000000

* **Đại số quan hệ:**

1. Cho mỗi phim mà 'BALIN MIREILLE' đã tham gia đóng phim, hãy cho biết tổng thù lao đã trả cho các diễn viên

* **SQL:**

select ho, ten, sum(thulao)

from dienvien d

JOIN film f ON d.idFilm = f.idFilm

JOIN canhan c ON d.idCN = c.idCN

where d.idFilm in (select f.idFilm

from dienvien d

JOIN canhan c ON d.idCN = c.idCN

JOIN film f ON f.idFilm = d.idFilm

where ho = 'RIVA' and ten = 'EMMANUELLE')

group by ho, ten

* **Đại số quan hệ:**

1. Tìm họ tên các diễn viên, họ tên người thực hiện của bộ phim 'RESULTADO FINAL'

* **SQL:**

select tua, ho || ' ' || ten as NguoiThucHien

from film f

JOIN canhan cn ON cn.idCN = f.nguoithuchien

where tua = 'RESULTADO FINAL';

* **Đại số quan hệ:**

1. Họ tên diễn viên được trả thù lao cao nhất từ một bộ phim nào đó

* **SQL:**

select d.idCN, ho, ten, tua, thulao as thulao\_caonhat

from dienvien d

JOIN canhan c ON d.idCN = c.idCN

JOIN film f ON f.idFilm = d.idFilm

where thulao = (select max(thulao) from dienvien)

* **Đại số quan hệ:**

1. Họ tên diễn viên có tổng thu lao thấp nhất

* **SQL:**

select d.idCN, ho, ten, sum(thulao) as TongThuLao

from dienvien d

JOIN canhan c ON d.idCN = c.idCN

group by d.idCN, ho, ten

having sum(thulao) = (select min(sum(thulao))

from dienvien group by idCN)

* **Đại số quan hệ:**

1. Tính số cá nhân đã tham gia đóng phim

* **SQL:**

select tua, count(\*) as So\_DienVien

from film f

JOIN dienvien d ON f.idFilm=d.idFilm

group by tua

* **Đại số quan hệ:**

1. Tính số cá nhân đã thực hiện phim

* **SQL:**

select nguoithuchien, count(\*) as so\_phim

from film f

JOIN canhan c ON f.nguoithuchien = c.idCN

group by nguoithuchien order by nguoithuchien ASC

* **Đại số quan hệ:**

1. Tính số cá nhân vừa đóng phim vừa thực hiện phim

* **SQL:**

select count(\*) as SoCaNhan\_DongVaThucHienPhim

from (select distinct idCN from dienvien

INTERSECT

select distinct nguoithuchien from film )

* **Đại số quan hệ:**

1. Tính số cá nhân chỉ đóng phim

* **SQL:**

select count(\*) as SoCaNhan\_Chi\_DongPhim

from (select distinct idCN

from dienvien

where idCN not in(select distinct nguoithuchien from film ))

* **Đại số quan hệ:**

1. Tính số cá nhân chỉ thực hiện phim

* **SQL:**

select count(\*) as SoCaNhan\_Chi\_THPhim

from (select distinct nguoithuchien from film )

* **Đại số quan hệ:**

1. Tính số cá nhân không đóng phim cũng không thực hiện phim

* **SQL:**

select count(\*) as SoCaNhan\_KhongLamGiCa

from (select distinct idCN

from canhan

where idCN not in (select distinct idCN from dienvien

union

select distinct nguoithuchien from film)

)

* **Đại số quan hệ:**

1. Những diễn viên nào đã tham gia ít nhất là một phim giống như diễn viên 'LANCASTER BURT'

* **SQL:**

select distinct ho, ten, tua

from dienvien d

JOIN film f ON d.idFilm = f.idFilm

JOIN canhan c ON d.idCN = c.idCN

where d.idFilm = ANY (select f.idFilm

from dienvien d

JOIN canhan c ON d.idCN = c.idCN

JOIN film f ON f.idFilm = d.idFilm

where ho = 'LANCASTER' and ten = 'BURT')

order by ho,ten

* **Đại số quan hệ:**

1. Những diễn viên nào đã tham gia ít nhất là tất cả các phim mà diễn viên 'RIVA EMMANUELLE' đã tham gia đóng phim

* **SQL:**

select distinct ho, ten, tua

from dienvien d

JOIN film f ON d.idFilm = f.idFilm

JOIN canhan c ON d.idCN = c.idCN

where d.idFilm = ALL (select f.idFilm

from dienvien d

JOIN canhan c ON d.idCN = c.idCN

JOIN film f ON f.idFilm = d.idFilm

where ho = 'RIVA' and ten = 'EMMANUELLE')

order by ho,ten

* **Đại số quan hệ:**

1. Ai đã tham gia tất cả các phim do 'BIVEL' thực hiện

* **SQL:**

select distinct tua, ho, ten, thulao

from film f

JOIN dienvien d ON f.idFilm = d.idFilm

JOIN canhan c ON d.idCN = c.idCN

where nguoithuchien = ALL(select distinct idCN

from film f

JOIN canhan c ON f.nguoithuchien = c.idCN

where ho = 'BIVEL');

* **Đại số quan hệ:**