Câu hỏi ôn tập chương 4

1. Ngôn ngữ SQL là gì?

* SQL (Structure Query Language) là ngôn ngữ truy vấn có cấu trúc.
* SQL hỗ trợ việc tạo, xóa, lưu trữ và truy xuất dữ liệu trong CSDL quan hệ.

1. Thứ tự viết các mệnh đề trong câu SQL?

SELECT -> FROM -> WHERE -> GROUP BY -> HAVING -> ORDER BY

1. Thứ tự thực hiện các mệnh đề trong câu SQL?

FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY -> Kết quả truy vấn

1. Cho Q(A, B, C). Viết câu truy vấn tương đương với biểu thức đại số quan hệ πA,B(Q)

Select distinct A, B

From Q

1. Cho Q(A, B, C, D). Dấu \* ở mệnh đề Select trong câu truy vấn SELECT \* FROM Q có nghĩa là gì?

Dấu \* này dùng để chọn tất cả các trường có trong quan hệ Q

1. Cho Q(A: int, B: int, C: int). Viết 2 câu truy vấn tương đương với biểu thức đại số quan hệ πA,B,C(σA>B(Q))

* Select \* From Q Where A > B
* Select A, B, C From Q Where A > B

1. Cho Q(A: int, B: int, C: int). Tìm các chỗ sai trong câu truy vấn

SELECT A.Q FROM Q WHERE A.Q > ‘5’

Câu đúng: SELECT Q.A FROM Q WHERE Q.A > ‘5’

1. Cho SV(MaSV: string, Hoten: string, MaKh:string, DiemTB:real). Câu truy vấn nào dưới đây thực hiện lập danh sách SV thuộc Khoa có MaKh là ‘50’ và có điểm tb > 8:
2. SELECT MaSV, Hoten FROM SV WHERE MaKh = 50, DiemTB >8
3. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’, DiemTB >8
4. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’∧ DiemTB >8
5. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’ and DiemTB >8
6. Cho SV(MaSV: string, Hoten: string, MaKh:string, DiemTB:real). Câu truy vấn nào dưới đây sai và tại sao?
7. SELECT MaSV, Hoten FROM SV WHERE MaKh = 50, DiemTB >8. Sai vì MaKh = 50 phải có ‘ ’ và giữa các biểu thức điều kiện không dùng dấu “,”. Câu đúng là:   
   SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’ and DiemTB >8
8. SELECT MaSV.SV, Hoten.SV FROM SV WHERE DiemTB >8. Sai vì trong lệnh SELECT tên quan hệ phải đứng trước tên cột. Câu đúng là:  
   SELECT SV.MaSV, SV.Hoten FROM SV WHERE DiemTB >8
9. SELECT SV.MaSV, SV.Hoten FROM SV WHERE DiemTB >8

1. Cho NV(MaNV: string, Hoten: string, Luong: Real, MaPB:string). Câu SQL nào dưới đây đúng:
2. SELECT MaNV, Hoten, Luong \* 2 From NV WHERE MAPB = ‘50’
3. SELECT MaNV, Hoten, Luong \* 2 Thuong From NV WHERE MAPB = ‘50’
4. SELECT MaNV, Hoten, Luong \* 2 as Thuong Form NV WHERE MAPB = ‘50’
5. Cho NV(MaNV: string, Hoten: string, Luong: Real, MaPB:string). Hãy cho biết kết quả của câu truy vấn: SELECT MaNV, Hoten, Luong \* 2 as Thuong FROM NV

Một quan hệ mới có 3 cột là MaNV, Hoten, Thuong và các dòng được lấy từ quan hệ NV trong đó Thuong = Luong \* 2.

1. Từ khóa AS trong câu truy vấn SELECT MaNV, Hoten, Luong \* 2 as Thuong FROM NV dùng để làm gì?

Từ khóa AS trong câu truy vấn trên dùng để đặt tên cột Lương \* 2 là Thuong.

1. Cho NhanVien(MaNV: string, Hoten: string, Luong: Real). câu truy vấn dưới thực hiện việc gì:

SELECT NV.MaNV, NV.Hoten, NV.Luong FROM NhanVien as NV WHERE Luong >1000

Câu truy vấn trên dùng để lập một quan hệ mới có tên là NV và có các thuộc tính giống với quan hệ NhanVien nhưng phải thỏa Luong > 1000

1. Từ khóa AS trong câu truy vấn SELECT MaNV, Hoten, Luong FROM NhanVien as NV dùng để làm gì?

Từ khóa AS trong câu truy vấn trên dùng để đổi tên NhanVien thanh NV.

1. bí danh của thuộc tính được đặt ở mệnh đề SELECT có được dùng ở các mệnh đề khác không? Tại sao?

Bí danh ở mệnh đề SELECT có thể sử dụng các mệnh đề khác vì đây là một tên tạm thời cho thuộc tính nên ta có thể sử dụng cho bất kỳ mệnh đề nào.

1. Phát biểu nào dưới đây đúng:
2. Các biểu thức điều kiện ở mệnh đề WHERE phải được liên kết bằng AND hoặc OR
3. Các biểu thức điều kiện ở mệnh đề WHERE phải được phân cách bằng dấu phẩy
4. LIKE hoặc NOT LIKE dùng để làm gì?

LIKE và NOT LIKE: dùng trong so sánh chuỗi.

1. Ý nghĩa của ký tự đại diện ‘%’ hoặc ‘\_’ trong biểu thức điều kiện so sánh chuỗi.

Các ký tự đại diện: ‘%’ (đại diện một nhóm ký tự bất kỳ), ‘\_’ (đại diện 1 ký tự bất kỳ).

1. Cho Q(A: nvarchar(4), B: int). câu SQL SELECT \* FROM Q WHERE A like ‘%a% thực hiện việc gì?

Liệt kê tất cả các thuộc tính có ký tự “a” ở cột A của quan hệ Q.

1. Cho Q(A: varchar(4), B: int). Câu SQL SELECT \* FROM Q WHERE A like ‘\_a% thực hiện việc gì?

Liệt kê tất cả các thuộc tính có ký tự “a” đứng thứ 2 ở cột A của quan hệ Q.

1. Cho Q(A: varchar(4), B: int). Viết câu SQL tương đương với câu   
    SELECT \* FROM Q WHERE B Between 10 and 20

SELECT \*

FROM Q

WHERE B>=10 and B<=20;

1. Hàm gộp là gì và gồm những hàm nào?

Hàm gộp là hàm nhận vào một tập giá trị của một thuộc tính trong một quan hệ và cho ra kết quả là một giá trị

Các hàm gộp là: MAX,MIN,AVG,SUM,COUNT

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Câu SQL SELECT count(\*), avg(luong) FROM NV thực hiện việc gì?

Sử dụng hàm gộp tính tổng số lượng nhân viên (tổng số lượng bộ có trong quan hệ) và tính mức lương trung bình của số nhân viên đó có trong quan hệ NV

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT count(\*), avg(luong) FROM NV là gì?

Một bảng gồm hai thuộc tính và 1 bộ: cột đầu tiên không có tên, cột thứ 2 tên luong lần lượt là số lượng nhân viên và lương trung bình trong quan hệ NV

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT count(\*) Sonv, avg(luong) as Luongtb FROM NV là gì?

Một bảng gồm hai thuộc tính và 1 bộ: cột đầu tiên tên Sonv, cột thứ hai là Luongtb lần lượt là tổng số nhân viên và mức lương trung bình của họ có trong quan hệ NV

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Câu SQL SELECT count(MaNV), avg(luong) FROM NV GROUP BY MaPB thực hiện việc gì?

Sử dụng hàm tính tổng số nhân viên và lương trung bình của nhóm nhân viên đó có cùng MaPB (gom nhóm theo MaPB) trong quan hệ NV

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT MaPB, count(MaNV), avg(luong) FROM NV GROUP BY MaPB là gì?

Gồm 3 cột và số bộ phụ thuộc vào số lượng MaPB đôi một khác nhau, các cột bao gồm MaPB, MaNV, luong lần lượt là mã phòng ban, tổng số lượng nhân viên và lương trung bình của họ theo MaPB trong quan hệ NV

1. Phát biểu nào dưới đây đúng khi câu SQL có sử dụng mệnh đề GROUP BY
2. Các thuộc tính có trong mệnh đề SELECT phải có trong mệnh đề GROUP BY hoặc trong hàm gộp
3. Thuộc tính có trong mệnh đề GROUP BY có thể không có trong mệnh đề SELECT
4. Phát biểu nào dưới đây đúng:
5. Trong câu SQL, mệnh đề HAVING được sử dụng khi điều kiện chọn liên quan tới hàm gộp
6. Trong câu SQL, mệnh đề HAVING (nếu có) phải nằm sau mệnh đề GROUP BY
7. Phát biểu nào dưới đây đúng:
8. Trong câu SQL, khi sử dụng mệnh đề HAVING thì phải có mệnh đề GROUP BY
9. Trong câu SQL, mệnh đề WHERE sử dụng với điều kiện không dùng hàm gộp
10. Trong câu SQL, có thể có cả mệnh đề WHERE và mệnh đề HAVING
11. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    câu SQL: SELECT MaPB, count(MaNV) FROM NV GROUP BY MaPB HAVING count(MaNV) > 2; thực hiện việc gì?

Từ quan hệ MaPB, gộp các giá trị theo MaPB, chọn ra các MaPB có MaNV>2, tiến hành chiếu các cột MaPB và dùng hàm gộp tính tổng số lượng bộ trong thuộc tính MaNV.

1. Cho Q(A: varchar(10), B: int, C: varchar(4)). Chỉ những chỗ sai của câu SQL:   
    SELECT A, count(b) as v FROM Q.A GROUP BY C HAVING v > 2;

* Thuộc tính b không được viết hoa như những gì có trong quan hệ Q.
* Q.A là thuộc tính chứ không phải là một quan hệ
* Do thuộc tính b không có nên việc đổi tên thành v ở HAVING so sánh với 2 là không thỏa

1. Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)). Chỉ những chỗ sai của câu SQL: SELECT MaPB, count(MaNV) as sonv FROM NV n WHERE sonv > 2;

* Trong hàm gộp nhưng dùng WHERE thay vì dùng HAVING

1. Mệnh đề ORDER BY dùng để làm gì?

Dùng để sắp xếp thứ tự của các bộ

1. Cho NV (MaNV, Hoten, NgSinh). Câu SQL:   
    SELECT Hoten, NgSinh FROM NV ORDER BY Hoten, NgSinh DESC; thực hiện việc gì?

Từ quan hệ NV, chiếu lấy các thuộc tính Hoten, NgSinh, sắp xếp các bộ theo thứ tự tăng dần của tên nếu có tên trùng nhau thì sắp xếp theo NgSinh giảm dần

1. Cho Q(A, B, C). Viết câu SQL cho kết quả tương đương với câu   
    SELECT \* FROM Q ORDER BY A DESC, C

SELECT \*

FORM Q

ORDER BY A DESC, C ASC

1. Trình bày phép kết nội và phép kết ngoài

* Kết nội là nhận các giá trị thỏa mãn điều kiện kết của hai quan hệ
* Kết ngoại: <qh1> và <qh2>
* Kết ngoại trái: bao gồm kết nội và những giá trị có trong <qh1> nhưng không có trong <qh2>, lúc này các thuộc tính trong <qh2> sẽ mang giá trị Null
* Kết ngoại phải: bao gồm kết nội và những giá trị có trong <qh2> nhưng không có trong <qh1>, lúc này các thuộc tính trong <qh1> sẽ mang giá trị Null
* Kết ngoại đầy đủ: bao gồm kết nội, những giá trị có trong <qh1> nhưng không có trong <qh2>, lúc này các thuộc tính trong <qh2> sẽ mang giá trị Null cùng đó những giá trị có trong <qh2> nhưng không có trong <qh1>, lúc này các thuộc tính trong <qh1> sẽ mang giá trị Null

1. Cho Q1(A: int, B: int) và Q2(C: int, D: int) và câu SQL   
   SELECT A, C FROM Q1 INNER JOIN Q2 ON A = C; thực hiện việc gì

Từ quan hệ Q1 kết với nội với Q2 với điều kiện kết là thuộc tính A = thuộc tính C, chiếu lấy cột A và C.

1. Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL nào dưới đây đúng
2. SELECT A, C FROM Q1 JOIN Q2 ON A = C;
3. SELECT A, C FROM Q1 INNER JOIN Q2 ON A = C;
4. Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL nào dưới đây đúng
5. SELECT A, C FROM Q1 LEFT JOIN Q2 ON A = C;
6. SELECT A, C FROM Q1 LEFT OUTER JOIN Q2 ON A = C;
7. Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL   
   SELECT \* FROM Q1 FULL JOIN Q2 ON A = C; thực hiện việc gì?

Thực hiện phép kết ngoài đầy đủ giữa A và C, tất cả các dòng ở A sẽ kết với các dòng ở C những dòng ở A không kết được với các dòng ở C thì sẽ mang giá trị null, tất cả các dòng ở C sẽ kết với các dòng ở A những dòng ở C không kết được với các dòng ở A thì sẽ mang giá trị null

1. Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL   
   SELECT \* FROM Q1 FULL OUTER JOIN Q2 ON A = C; tương đương với
2. SELECT \* FROM Q1 JOIN Q2 ON A = C;
3. SELECT \* FROM Q1 FULL JOIN Q2 ON A = C
4. Cho Q1(A: int, B: int) và Q2(C: int, D: char(5)). Câu SQL   
   SELECT \* FROM Q1 OUTER JOIN Q2 A = C WHERE D = null; sai ở những chỗ nào?

Sửa: SELECT \* FROM Q1 LEFT/RIGHT/FULL OUTER JOIN Q2 ON A = C WHERE D is null

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL  
   (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky); thực hiện việc gì?

(Lấy cột MaSV và tất cả các dòng trong bảng SV) ngoại trừ (Lấy cột MaSV và tất cả các dòng trong bảng Dangky)

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới tương đương với câu truy vấn (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)
2. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV
3. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH IS NULL
4. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH = NULL
5. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới tương đương với câu truy vấn (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)
6. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH IS NULL
7. SELECT s.MaSV FROM SV s LEFT JOIN Dangky d ON s.MaSV = d.MaSV WHERE Dangky.MaMH IS NULL
8. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Chỉ những chỗ sai trong câu SQL  
   SELECT MaSV FROM SV OUTER JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH = NULL;

Sửa: SELECT MaSV FROM SV LEFT/RIGHT/FULL OUTER JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH is NULL

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới cho kết quả tương đương với câu truy vấn (SELECT MaSV FROM SV) INTERSECT (SELECT MaSV FROM Dangky)
2. SELECT MaSV FROM Dangky
3. (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)
4. SELECT MaSV FROM SV
5. Cho Q1(A, B) và Q2(A, B). Câu SQL  
   (SELECT A, B FROM Q1) INTERSECT (SELECT A, B FROM Q2); thực hiện việc gì?

(Lấy cột A, B và tất cả các dòng trong bảng Q1) giao nhau (Lấy cột A, B và tất cả các dòng trong bảng Q2)

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL dưới thực hiện việc gì (SELECT MaSV FROM SV) UNION (SELECT MaSV FROM Dangky)

(Lấy cột MaSV và tất cả các dòng trong bảng SV) hội (Lấy cột MaSV và tất cả các dòng trong bảng Dangky)

1. Cho Q1(A, B) và Q2(A, B). Câu SQL  
   (SELECT \* FROM Q1) UNION (SELECT \* FROM Q2); thực hiện việc gì?

(Lấy cột A, B và tất cả các dòng trong bảng Q1) hội (Lấy cột A, B và tất cả các dòng trong bảng Q2)

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu truy vấn  
    SELECT Hoten FROM SV WHERE MaSV NOT IN (SELECT MaSV FROM Dangky); thực hiện việc gì?

Lấy cột MaSV và tất cả các dòng trong bảng Dangky sau đó tìm Hoten sinh viên nào mà MaSV không tương ứng với một trong các MaSV trong câu truy vấn Lấy cột MaSV và tất cả các dòng trong bảng Dangky

1. Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL tương đương với câu:  
    SELECT MaSV FROM SV WHERE MaSV NOT IN (SELECT MaSV FROM Dangky)

SELECT MaSV

FROM SV sv, (SELECT MaSV

FROM Dangky) Q

WHERE Q.MaMH is NULL

;

1. Cho NV(MaNV, Hoten, Luong). Câu SQL   
    SELECT \* FROM NV WHERE Luong > (SELECT AVG(Luong) FROM NV); thực hiện việc gì?
2. Liệt kê danh sách NV có lương cao nhất
3. Liệt kê danh sách NV có lương lớn hơn mức lương trung bình
4. Cho NV(MaNV, Hoten, Luong). Câu SQL nào dưới đây tương đương với  
   SELECT \* FROM NV WHERE Luong > (SELECT AVG(Luong) FROM NV);
5. SELECT \* FROM NV, (SELECT AVG(Luong) as ltb FROM NV) Q WHERE Luong > LTB
6. SELECT \* FROM NV HAVING Luong > (SELECT AVG(Luong) FROM NV)
7. Cho NV(MaNV, Hoten, Luong). Câu SQL   
    SELECT \* FROM NV WHERE Luong >= all (SELECT Luong FROM NV); thực hiện việc gì?

Cho ra kết quả là 1 quan hệ có 1 cột duy nhất đó là cột Luong và giá trị trong cột Luong la Luong cua NV sau đó lấy ra MaNV, Hoten, Luong mà Luong lớn hơn hoặc bằng trong tất cả các Luong trong bảng NV

1. Cho NV(MaNV, Hoten, Luong). Câu SQL   
   SELECT \* FROM NV WHERE Luong = (SELECT max(Luong) FROM NV); thực hiện việc gì?

Cho ra kết quả là 1 giá trị đó là t Luong lớn nhất toàn bộ NV sau đó lấy ra MaNV, Hoten, Luong mà Luong bằng với mức lương lớn nhất trong bảng NV

1. Cho NV(MaNV, Hoten, Luong). Câu SQL nào dưới đây tương đương với  
   SELECT \* FROM NV WHERE Luong >= all (SELECT Luong FROM NV)
2. SELECT \* FROM NV WHERE Luong >= any (SELECT Luong FROM NV);
3. SELECT \* FROM NV WHERE Luong = (SELECT max(Luong) FROM NV)
4. Truy vấn con tương quan là gì?

Cho phép so sánh giữa thuộc tính của quan hệ bên trong truy vấn con với thuộc tính của quan hệ trong truy vấn bao bên ngoài

1. Công dụng của EXISTS

được dùng để kiểm tra kết quả của câu truy vấn con tương quan là rỗng hay không rỗng. Nếu kết quả của câu truy vấn là rỗng thì EXISTS trả về False, ngược lại nó trả về True.

1. Cho NV(MaNV, Hoten) và ThanNhan(MaNV, TenThannhan). Câu SQL nào dưới đây đúng
2. SELECT Hoten FROM NV WHERE EXISTS (SELECT \* FROM THANNHAN tn WHERE NV.MaNV = tn.MaNV);
3. SELECT Hoten FROM NV WHERE MaNV EXISTS (SELECT \* FROM THANNHAN tn WHERE NV.MaNV = tn.MaNV);