Câu hỏi ôn tập chương 4

1. Ngôn ngữ SQL là gì?

SQL là để ngôn ngữ truy vấn cơ sở dữ liệu

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

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?

* Chiếu lên tất cả các thuộc tính của 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 distinct \*
* 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’

* Q.A, A có kiểu dữ liệu là int k thể so sánh với chuỗi ‘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 => and hoặc ^
8. SELECT MaSV.SV, Hoten.SV FROM SV WHERE DiemTB >8 => cú pháp (Tên quan hệ. Thuộc tính ) SV.MS, SV.Hoten
9. SELECT SV.MaSV, SV.Hoten FROM SV WHERE DiemTB >8
10. Cho NV(MaNV: string, Hoten: string, Luong: Real, MaPB:string). Câu SQL nào dưới đây đúng:
11. SELECT MaNV, Hoten, Luong \* 2 From NV WHERE MAPB = ‘50’
12. SELECT MaNV, Hoten, Luong \* 2 Thuong From NV WHERE MAPB = ‘50’
13. SELECT MaNV, Hoten, Luong \* 2 as Thuong Form NV WHERE MAPB = ‘50’
14. 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

* Kết quả là 3 cột MaNV, Hoten, Thuong với Thuong = Luong \*2

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

* As dùng đặt bí danh của thuộc tính có Luong \* 2

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

* Thực hiện chọn ra những nhân viên có lương lớn hơn 1000

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

* AS dùng để đổi tên quan hệ NhanVien thành NV

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

* Không, vì theo thứ tự thực hiện From-> Where-> Group by -> Having -> Select -> Order by các mệnh đề có thứ tự thực hiện trước Select sẽ bị lỗi.

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?

* Dùng để so sánh chuỗi, kết hợp với % đại diện cho 1 nhóm kí tự hoặc \_ đại diện cho 1 kí tự

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

* % đại diện 1 nhóm kí tự, \_ đại diện 1 kí tự

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

* Thực hiện việc tìm ra những giá trị trong cột A có kí tự a xuất hiện bất kì vị trí nào.

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

* Thực hiện việc tìm ra những giá trị trong cột A có kí tự a xuất hiện ở vị trí thứ 2.

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 A,B
* FROM Q
* WHERE B>=10 and B<= 20

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

* Hàm gộp nhận vào 1 tập giá trị của 1 cột trả về 1 giá trị
* Hàm gộp gồm sum,max,count,min,avg

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

* Thực hiện việc đếm số lượng nhân viên và tính trung bình lương của tất cả nhân viên

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?

* Kết quả là 2 cột 1 bộ: cột 1 hiển thị số lượng nhân viên, cột 2 hiển thị luong trung bình của all nhân viên

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?

* Kết quả là 2 cột 1 bộ: cột 1 Sonv hiển thị số lượng nhân viên, cột 2 Luongtb hiển thị lương trung bình của all nhân viên

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?

* Thực hiện việc gom nhóm MaPB của quan hệ NV sau đó đếm số lượng nhân viên ở mỗi phòng và tính trung bình lương nhân viên của mỗi phòng.

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?

* Kết quả là 3 cột MaPB sau khi đã gom nhóm, số lượng nhân viên của mỗi phòng ban, trung bình lương của tất cả nhân viên ở mỗi phong ban.

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?

* Gom nhóm MaPB, đếm số lương nhân viên mỗi phòng lớn hơn 2, chiếu lên MaPB và số lượng nhân viên

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;

* Having được thực hiện trước hơn Select nên không thể sử dụng thuộc tính v được.

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;

* Where được thực hiện trước hơn SELECT nên không thể sử dụng thuộc tính sonv được.

1. Mệnh đề ORDER BY dùng để làm gì?

* Để sắp xếp, mặc định asc tăng dần, decs giảm dần

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

* Sắp xếp Hoten tăng dần, nếu trùng thì sắp xếp ngày sinh theo thứ tự 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 \*
* FROM Q
* ORDER BY A DESC, C ASC

1. Trình bày phép kết nội và phép kết ngoài

Phép kết nội:

C1:

* SELECT <tên cột 1> , <tên cột 2>,…
* FROM <quan hệ 1>,<quan hệ 2>
* Where <điều kiện kết>

C2:

* SELECT <tên cột 1>,<tên cột 2>,…
* FROM <quan hệ 1> inner join <quan hệ 2> on <điều kiện kết>

Phép kết ngoại:

* SELECT <tên cột 1>, <tên cột 2>,…
* FROM <quan hệ 1> Left/Right/Full Outer join <quan hệ 2>
* On <điều kiện kết>

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?

* Kết nội 2 quan hệ Q1, Q2 với điều kiện A=C cho ra kết quả 2 cột A,C giống nhau

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 việc kết đẩy đủ 2 quan hệ Q1,Q2 với điều kiện A=C

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?

* OUTER JOIN không rõ là kết trái, kết phải hay kết đầy đủ. D is null mới chọn ra những

Giá trị trong quan hệ D là null còn D = null sẽ cho kết quả là rỗng.

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

* Thực hiện việc chọn ra những sinh viên chưa đăng kí môn học.

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;

* Outer join không rõ là kết trái, phải , đầy đủ. 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?

* Thực hiện việc chọn ra những bộ có trong Q1.A,Q2.B và Q2.A,Q2.B

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)

* Thực hiện việc lấy tất cả những MaSV có trong SV hoặc Dangky

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

* Thực hiện việc lấy tất cả A,B có trong Q1 hoặc 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?

* Chọn ra những sinh viên chưa đăng kí môn học.

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 EXCEPT (SELECT MASV FROM Dangky\_
* SELECT SV.MASV
* FROM SV left outer join Dangky
* On SV.MASV = Dangky.MASV
* Where Dangky.MASV 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?

* Thực hiện việc in những nhân việc có lương cao nhất

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

* Thực hiện việc in ra những nhân việc có lương cao nhất

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 tru 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 trả về False, ngược lại 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);