

HW #1 - Discussion

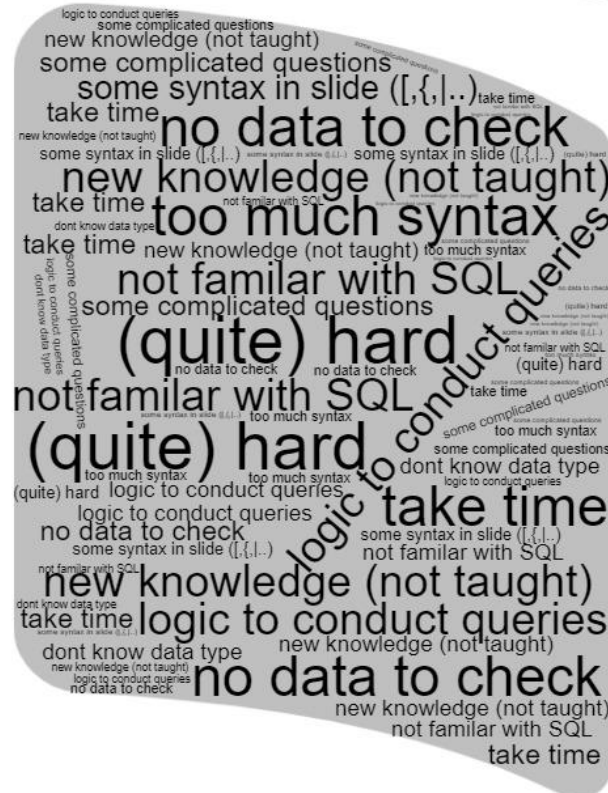
Vũ Tuyết Trinh

trinhvt@soict.hust.edu.vn

Summary

- Turn-in
 - On time (+1.. 4 min) : 86
 - Late (11..21 hours): 2
 - Submitted but not turn-in: 3
- Viewed only : 8
- To discuss
 - 1: f g i j h
 - 2: d h f g (c j)

Difficulties



SQL Syntax (from slides3_SQL(part2).pdf)

SELECT [all|distinct]

[xxx] not obligatory

{*|{table_name.*|expr[alias]}|view_name.*}

{xxx} expression

[,{table_name.*|expr[alias]}]...

FROM table_name [alias][,table_name[alias]] ...

[WHERE condition]

[GROUP BY expr [,expr] ...]

[HAVING condition]

[{UNION|UNION ALL|INTERSECT|MINUS}

SELECT ...]

A|B ~ may be A or B

[ORDER BY {expr|position} [ASC|DESC]

[,expr|position} [ASC|DESC]

Exercise 1

LECTURER(StaffNO, Title, FName, LName, Address, Salary, ResearchArea)

STUDENT(StudentNO, FName, LName, DOB, Address, StudentStatus)
(StudentStatus can be 'C' (current) or 'G' (graduated))

SUBJECT(SubjectCode, Name, Prerequisite, Credit)

LECTURE(LectureCode, SubjectCode, StaffNO, Semester, LectDay, LectTime, Venue)
(SubjectCode and StaffNO are foreign keys)

STUDENT_ENROLMENT(LectureCode, StudentNO)
(LectureCode and StudentNO are foreign keys)

f

LECTURER(StaffNO, Title, FName, LName, Address, Salary, ResearchArea)

STUDENT(StudentNO, FName, LName, DOB, Address, StudentStatus)
(StudentStatus can be 'C' (current) or 'G' (graduated))

SUBJECT(SubjectCode, Name, Prerequisite, Credit)

LECTURE(LectureCode, SubjectCode, StaffNO, Semester, LectDay, LectTime, Venue)
(SubjectCode and StaffNO are foreign keys)

STUDENT_ENROLMENT(LectureCode, StudentNO)
(LectureCode and StudentNO are foreign keys)

```
SELECT ST.StudenNo, ST.LName, ST.FName, ST.DOB, ST.StudentStatus
FROM STUDENT ST , SUBJECT SU, Student_ENROLMENT SE, LECTURE LE
WHERE ST.StudentNO = SE.StudenNO AND
SE.LectureCode = LE.LectureCode AND
LE.SubjectCode = SU.SubjectCode AND
SU.Name = 'Database'
ORDER BY ST.LName;
```

g

```
SELECT StaffNO, FName, LName
FROM LECTURER L
WHERE NOT EXISTS
(SELECT StaffNO FROM LECTURE WHERE L.StaffNO = LECTURE.StaffNO);
```

```
SELECT StaffNO, FName, LName
FROM LECTURER
WHERE StaffNO NOT IN (SELECT distinct StaffNO
FROM LECTURE );
```

LECTURER(StaffNO, Title, FName, LName, Address, Salary, ResearchArea)

STUDENT(StudentNO, FName, LName, DOB, Address, StudentStatus)
(StudentStatus can be 'C' (current) or 'G' (graduated))

SUBJECT(SubjectCode, Name, Prerequisite, Credit)

LECTURE(LectureCode, SubjectCode, StaffNO, Semester, LectDay, LectTime, Venue)

(SubjectCode and StaffNO are foreign keys)

STUDENT_ENROLMENT(LectureCode, StudentNO)
(LectureCode and StudentNO are foreign keys)

LECTURER(StaffNO, FName, LName, Address, Salary, ResearchArea)
 (StaffNO is the primary key)

STUDENT(StudentNO, FName, LName, DOB, Address, StudentStatus)
 (StudentStatus can be 'C' (current) or 'G' (graduated))

SUBJECT(SubjectCode, Name, Prerequisite, Credit)

LECTURE(LectureCode, SubjectCode, StaffNO, Semester, LectDay, LectTime, Venue)
 (SubjectCode and StaffNO are foreign keys)

STUDENT_ENROLMENT(LectureCode, StudentNO)
 (LectureCode and StudentNO are foreign keys)

```
SELECT LECTURER.StaffNO, FName, LName
FROM dbo.LECTURER AS L
WHERE (SELECT COUNT(SubjectCode) FROM dbo.LECTURE WHERE StaffNO = L.StaffNO) >= 3;
```

```
SELECT COUNT(SubjectCode)
```

```
FROM dbo.LECTURE WHERE StaffNO = "007"
```

```
GROUP BY StaffNO → result is collection (of 1 record) NOT a value
```

j

LECTURER(StaffNO, Title, FName, LName, Address, Salary, ResearchArea)

STUDENT(StudentNO, FName, LName, DOB, Address, StudentStatus)
(StudentStatus can be 'C' (current) or 'G' (graduated))

SUBJECT(SubjectCode, Name, Prerequisite, Credit)

LECTURE(LectureCode, SubjectCode, StaffNO, Semester, LectDay, LectTime, Venue)
(SubjectCode and StaffNO are foreign keys)

STUDENT_ENROLMENT(LectureCode, StudentNO)
(LectureCode and StudentNO are foreign keys)

```
SELECT SubjectCode, StaffNo, count(StudentNo)
FROM    LECTURE as L, STUDENT_ENROLMENT as SE
WHERE   L.LectureCode = SE.LectureCode
GROUP BY SubjectCode, StaffNo
```

```
SELECT SubjectCode, StaffNo, count(StudentNo)
FROM    LECTURE as L, STUDENT_ENROLMENT as SE
WHERE   L.LectureCode = SE.LectureCode
GROUP BY SE.LectureCode
```

Should avoid this !

Hotel (Hotel_No, Name, Address)

Room (Room_No, *Hotel_No*, Type, Price)

Booking (*Hotel_No*, *Guest_No*, Date_From, Date_To, *Room_No*)

Guest (Guest_No, Name, Address)

Underlying attributes are primary keys. Italic attributes are foreign key.

Date_From < Date_To

d

Hotel (Hotel_No, Name, Address)

Room (Room_No, *Hotel_No*, Type, Price)

Booking (*Hotel_No*, *Guest_No*, Date_From, Date_To, *Room_No*)

Guest (Guest_No, Name, Address)

Underlying attributes are primary keys. Italic attributes are foreign key.

Date_From < Date_To

```
Select      *
From        Room R
Where Room_No not in
      (select *
      from    Booking
      where Room_No = R.Room_No and
            ((Date_To <= '1/6/2016' and Date_From >='1/6/2016') or
             (Date_To <= '30/6/2016' and Date_From >='30/6/2016') or
             (Date_To between '1/6/2016' and '30/6/2016' and
              Date_From between '1/6/2016' and '30/6/2016')) )
```

f

Hotel (Hotel_No, Name, Address)

Room (Room_No, *Hotel_No*, Type, Price)

Booking (*Hotel_No*, *Guest_No*, Date_From, Date_To, *Room_No*)

Guest (Guest_No, Name, Address)

Underlying attributes are primary keys. Italic attributes are foreign key.

Date_From < Date_To

```
Select      count(Guest_No)
From        Booking
Where       (month (Date_From) <= 8 and month(Date_To) >=8
            and year(Date_From) <=2016 and year(Date_To) >=2016)
            or (year (Date_From) <2016 and year(Date_To) >2016)
```

9

Hotel (Hotel_No, Name, Address)

Room (Room_No, *Hotel_No*, Type, Price)

Booking (*Hotel_No*, *Guest_No*, Date_From, Date_To, *Room_No*)

Guest (*Guest_No*, Name, Address)

Underlying attributes are primary keys. Italic attributes are foreign key.

Date_From < Date_To

```
SELECT Guest.*
```

```
FROM (Guest join Booking on Guest.Guest_No = Booking.Guest_No)
```

```
JOIN Hotel ON Booking.Hotel_No = Hotel.Hotel_No
```

```
WHERE (SELECT NOW() BETWEEN Date_From AND Date_To)
```

```
AND Hotel.Name = 'GrosvenorHotel';
```