

Comments for TMA03

Semester OCT 2003, by Andy Au (t420135), Group 4

This document is aimed at providing readers a guideline for addressing the key points of each question in TMA03, whereas the most common errors made by colleague.

Q1.

Students may practice the SQL using Infomaker and get hints. It is not uncommon to see answers for a) says `student_id` & `course_code` should be group by. This is a rule, a justification to the errors, but not pointing out the problem that `AVG` return 1 row while others return many. For part b) Where come before Group By, syntax! For part c) pls give what the query ask for in predicate English; cases where answer in form of statement translation of SQL will lose marks.

Q2.

- ai) note to select **DISTINCT** `staff_no`, `name`, `region` from (JOIN `staff` and `enrolment` on common `staff_no` and `tutor_no`)
- aii) subquery: select from `staff` where `staff_no` NOT IN (select `tutor_no` from `enrolment` where `tutor_no` is not null) or
outer join: select ... from `staff` left outer join `enrolment` on `staff_no` = `tutor_no` where `tutor_no` is null
- aiii) note to select ... `CAST(AVG(grade) AS Decimal(5,2))`, join `student` and `assignment` on `student_id`, group by `region` and use `Having` to restrict `AVG(grade)>50`
- aiv) select **DISTINCT** ... from `student` `s`, `enrolment` `e`, `staff` `t` WHERE `s.counsellor_no` = `t.staff_no`, `t.staff_no` = `e.tutor_no`
and also `e.student_id=s.student_id`
- bi) `course_code`, `course_title` and total **number of students (some students missed this)** enrolled on each course on which >3 students enrolled
- bii) Most students identified the `From`, `Where`... logical processing & the order and intermediate table. I found some cases not to point out it is a **Cartesian product** of `course` and `enrolment` in `FROM` clause.

Q3.

First of all, I would say many students score high in this question. I highlight the most common errors in blue:

- a) create view... select ... from `student` where `region` = '4' (with quote ', it is CHAR)
- b) create table `Student_id` **CHAR(3)**, `name` **VARCHAR(12)**, `registered` **SMALLINT**
- ci) grant select... to public
- cii) grant insert, delete update ... to faculty with grant option; I ever see an answer giving 3 grant statement, there's OK but the problem is only one with grant option
- ciii) note the last Rows should allow **Only Reg 4**
- di) create table... `student_id` **CHAR(3)**, `course_code` **CHAR(2)**, Primary Key (`student_id`, `course_code`)
- dii) insert into `r4_support` values ('s10', 'c3') ... straightforward
- diii) Alter table `r4_support` add foreign key (`student_id`, `course_code`) references `enrolment` on delete cascade, majority score!
- e) i) and iii) insert the row but ii) the key point is **referential constraint violation, s10 is not enrolled on course c5**
- eiv) change the name of `R4_student` view and the base table `student`, (some people not mention the change is also reflected in **R4_student view and say it contains no data for update**); `R4_student_orig` no change
- f) `R4_support` (`region` 4 only) – Alter table `R4_support` add check (`student_id` in (select `student_id` from `r4_student`))
- gi) alter table... add `city` `varchar(15)`
- gii) update ...set `city` = 'London' where `student_id` = 's10'