The following questions are taken from the appendix section of a book called *SQL cookbook* written by Anthony Molinario. These questions and solutions were first covered in *The Essence of SQL* written by David Rozenshtein.Anthony Molinario said,“ *The Essence of SQL* was the best book ever written on SQL.” and included some the questions in his book. I also found this section to be very intriguing to a SQL zealot like me. I will post the questions and the solutions from each author and also my very own solutions.

The following tables are based on Rozenshtein’s book and will be used in this appendix:

**create** **table** students

**(**

sno **integer,**

sname **varchar(**10**),**

age **integer**

**);**

**create** **table** courses

**(**

cno **varchar(**5**),**

title **varchar(**10**),**

credits **integer**

**);**

**create** **table** professors

**(**

lname **varchar(**10**),**

dept **varchar(**10**),**

salary **integer,**

age **integer**

**);**

**create** **table** take

**(**

sno **integer,**

cno **varchar(**5**)**

**);**

**create** **table** teach

**(**

lname **varchar(**10**),**

cno **varchar(**5**)**

**);**

**insert** **into** student **values** **(**1**,** 'AARON'**,** 20**)**

**insert** **into** student **values** **(**2**,** 'CHUCK'**,** 21**)**

**insert** **into** student **values** **(**3**,** 'DOUG'**,** 20**)**

**insert** **into** student **values** **(**4**,** 'MAGGIE'**,** 19**)**

**insert** **into** student **values** **(**5**,** 'STEVE'**,** 22**)**

**insert** **into** student **values** **(**6**,** 'JING'**,** 18**)**

**insert** **into** student **values** **(**7**,** 'BRIAN'**,** 21**)**

**insert** **into** student **values** **(**8**,** 'KAY'**,** 20**)**

**insert** **into** student **values** **(**9**,** 'GILLIAN'**,** 20**)**

**insert** **into** student **values** **(**10**,** 'CHAD'**,** 21**)**

**insert** **into** courses **values(**'CS112'**,** 'PHYSICS'**,** 4**)**

**insert** **into** courses **values(**'CS113'**,** 'CALCULUS'**,** 4**)**

**insert** **into** courses **values(**'CS114'**,** 'HISTORY'**,** 4**)**

**insert** **into** professor **values(**'CHOI'**,** 'SCIENCE'**,** 400**,** 45**)**

**insert** **into** professor **values(**'GUNN'**,** 'HISTORY'**,** 300**,** 60**)**

**insert** **into** professor **values(**'MAYER'**,** 'SCIENCE'**,** 400**,** 55**)**

**insert** **into** professor **values(**'POMEL'**,** 'SCIENCE'**,** 500**,** 65**)**

**insert** **into** professor **values(**'FEUER'**,** 'SCIENCE'**,** 400**,** 40**)**

**insert** **into** take **values** **(**1**,** 'CS112'**)**

**insert** **into** take **values** **(**1**,** 'CS113'**)**

**insert** **into** take **values** **(**1**,** 'CS114'**)**

**insert** **into** take **values** **(**2**,** 'CS112'**)**

**insert** **into** take **values** **(**3**,** 'CS112'**)**

**insert** **into** take **values** **(**3**,** 'CS114'**)**

**insert** **into** take **values** **(**4**,** 'CS112'**)**

**insert** **into** take **values** **(**4**,** 'CS113'**)**

**insert** **into** take **values** **(**5**,** 'CS113'**)**

**insert** **into** take **values** **(**6**,** 'CS113'**)**

**insert** **into** take **values** **(**6**,** 'CS114'**)**

**insert** **into** teach **values(**'CHOI'**,** 'CS112'**)**

**insert** **into** teach **values(**'CHOI'**,** 'CS113'**)**

**insert** **into** teach **values(**'CHOI'**,** 'CS114'**)**

**insert** **into** teach **values(**'POMEL'**,** 'CS113'**)**

**insert** **into** teach **values(**'MAYER'**,** 'CS112'**)**

**insert** **into** teach **values(**'MAYER'**,** 'CS114'**)**

Let’s begin!

**Questions:**

1. You want to find students who do not take ‘CS112’. Your query should return only Steve, Jing, Brian, Kay, Gillian, and Chad.
2. You want to find students who take ‘CS112’ or ‘CS114’ but NOT both. Your query should return Chuck, Maggie, and Jing.
3. You want to find students who take ‘CS112’ and NO other courses. Your query should return Chuck.
4. You want to find the students who take AT MOST two courses. Students who do not take any courses should be excluded. Your query should return Chuck, Doug, Maggie, Steve, and Jing.
5. You want to find students who are older than AT MOST two other students. Your query should return Jing, Maggie, Aaron, Gillian, Kay, and Doug.
6. You want to find students who take AT LEAST two courses. Your query should return Aaron, Doug, Maggie, and Jing.
7. You want to find students who take both ‘CS112’ and ‘CS114’. The students may take other courses, but they must take ‘CS112’ and ‘CS114’ as well. Your query should return Aaron and Doug.
8. Find students who are older than AT LEAST two other students. Your query should return Aaron, Chuck, Doug, Steve, Brian, Kay, Gillian, and Chad.
9. Find professors who teach exactly one course. Your query should return Pomel.
10. You want to find students who take ONLY ‘CS112’ and ‘CS114’. Your query should return Doug.
11. You want to find students who are older than EXACTLY two other students. Your query should return Aaron, Doug, Kay, and Gillian.
12. You want to find students who take all courses. Your query should return Aaron.
13. Find students who are older than any other students. Your query should return Steve.

**Solutions:**

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s **left** **join** take t

**on** **(**s**.**sno **=** t**.**sno**)**

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **max(case** **when** t**.**cno **=** 'CS112'

**then** 1 **else** 0 **end)** **=** 0

David Rozenshtein:

**select** **\***

**from** student

**where** sno **not** **in** **(select** sno

**from** take

**where** cno **=** 'CS112'

My solution is the same as David Rozenshtein’s.

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **sum(case** **when** t**.**cno **in** **(**'CS112'**,** 'CS114'**)**

**then** 1 **else** 0 **end)** **=** 1

David Rozenshtein:

**select** **\***

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**and** t**.**cno **in** **(**'CS112'**,** 'CS114'**)**

**and** s**.**sno **not** **in** **(select** a**.**sno

**from** take a**,** take **by**

**where** a**.**sno **=** b**.**sno

**and** a**.**cno **=** 'CS112'

**and** b**.**cno **=** 'CS114'**)**

Mine:

**select** s**.\***

**from** students s**,**

**(**

**select** sno**,** **count(**cno**)**

**from** take

**where** cno **IN** **(**'CS112'**,**'CS114'**)**

**group** **by** sno

**having** **count(**cno**)** **=** 1

**)** x

**where** s**.**sno **=** x**.**sno**;**

1. Anthony Molinario:

**select** s**.\***

**from** student s**,** take t1**,**

**(**

**select** sno

**from** take

**group** **by** sno

**having** **count(\*)** **=** 1

**)** t2

**where** s**.**sno **=** t1**.**sno

**and** t1**.**sno **=** t2**.**sno

**and** t1**.**cno **=** 'CS112'

David Rozenshtein:

**select** s**.\***

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**and** s**.**sno **not** **in** **(select** sno

**from** take

**where** cno **!=** 'CS112'**)**

My solution is the same as David Rozenshtein’s.

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **count(\*)** **<=** 2

David Rozenshtein:

**select** **distinct** s**.\***

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**and** s**.**sno **not** **in** **(select** t1**.**sno

**from** take t1**,** take t2**,** take t3

**where** t1**.**sno **=** t2**.**sno

**and** t2**.**sno **=** t3**.**sno

**and** t1**.**cno **<** t2**.**cno

**and** t2**.**cno **<** t3**.**cno

Mine:

**select** s**.**sname

**from** students s**,**

**(**

**select** sno**,** **count(**sno**)**

**from** take

**group** **by** sno

**having** **count(**sno**)** **<=** 2

**and** **count(**sno**)** **>** 0

**)** x

**where** s**.**sno **=** x**.**sno**;**

1. Anthony Molinario:

**select** s1**.\***

**from** student s1

**where** 2 **>=** **(select** **count(\*)**

**from** student s2

**where** s2**.**age **<** s1**.**age**)**

David Rozenshtein:

**select** **\***

**from** student

**where** sno **not** **in** **(**

**select** s1**.**sno

**from** student s1**,**

student s2**,**

student s3**,**

student s4

**where** s1**.**age **>** s2**.**age

**and** s2**.**age **>** s3**.**age

**and** s3**.**age **>** s4**.**age

**)**

Mine:

**select** **\***

**from** students

**where** sname **not** **in** **(**

**select** a**.**sname

**from** students a**,** students b

**where** a**.**age **-** b**.**age **>=** 3

**)**

**order** **by** 3**;**

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **count(\*)** **>=** 2

David Rozenshtein:

**select** **\***

**from** student

**where** sno **in** **(**

**select** t1**.**sno

**from** take t1**,**

take t2

**where** t1**.**sno **=** t2**.**sno

**and** t1**.**cno **>** t2**.**cno

Mine:

**select** s**.**sname

**from** students s**,**

**(**

**select** sno**,** **count(**sno**)**

**from** take

**group** **by** sno

**having** **count(**sno**)** **>=** 2

**)** x

**where** s**.**sno **=** x**.**sno**;**

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**and** t**.**cno **in** **(**'CS114'**,** 'CS112'**)**

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **min(**t**.**cno**)** **!=** **max(**t**.**cno**)**

David Rozenshtein:

**select** s**.\***

**from** student s**,**

take t1**,**

take t2

**where** s**.**sno **=** t1**.**sno

**and** t1**.**sno **=** t2**.**sno

**and** t1**.**cno **=** 'CS112'

**and** t2**.**cno **=** 'CS114'

Mine:

**select** s**.**sname

**from** students s**,**

**(**

**select** sno**,** **count(**cno**)**

**from** take

**where** cno **IN** **(**'CS112'**,**'CS114'**)**

**group** **by** sno

**having** **count(**cno**)** **=** 2

**)** x

**where** s**.**sno **=** x**.**sno**;**

1. Anthony Molinario:

**select** s1**.\***

**from** student s1

**where** 2 **<=** **(select** **count(\*)**

**from** student s2

**where** s2**.**age **<** s1**.**age**)**

David Rozenshtein:

**select** **distinct** s1**.\***

**from** student s1**,**

student s2**,**

student s3

**where** s1**.**age **>** s2**.**age

**and** s2**.**age **>** s3**.**age

Mine:

**select** **\***

**from** students

**where** sname **in** **(**

**select** a**.**sname

**from** students a**,** students b

**where** a**.**age **-** b**.**age **>=** 2

**)**

**order** **by** 3**;**

1. Anthony Molinario:

**select** p**.**lname**,** p**.**dept**,** p**.**salary**,** p**.**age

**from** professor p**,** teach t

**where** p**.**lname **=** t**.**lname

**group** **by** p**.**lname**,** p**.**dept**,** p**.**salary**,** p**.**age

**having** **count(\*)** **=** 1

David Rozenshtein:

**select** p**.\***

**from** professor p**,**

teach t

**where** p**.**lname **=** t**.**lname

**and** p**.**lname **not** **in** **(**

**select** t1**.**lname

**from** teach t1**,**

teach t2

**where** t1**.**lname **=** t2**.**lname

**and** t1**.**cno **>** t2**.**cno

Mine:

**select** lname**,** **count(**cno**)**

**from** teach

**group** **by** lname

**having** **count(**cno**)** **=** 1**;**

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**group** **by** s**.**sno**,** s**.**sname s**.**age

**having** **count(\*)** **=** 2

**and** **max(case** **when** cno **=** 'CS112' **then** 1 **else** 0 **end)** **+**

**max(case** **when** cno **=** 'CS114' **then** 1 **else** 0 **end)** **=** 2

David Rozenshtein:

**select** s1**.\***

**from** student s1**,** take t1**,** take t2

**where** s1**.**sno **=** t1**.**sno

**and** s1**.**sno **=** t2**.**sno

**and** t1**.**cno **=** 'CS112'

**and** t2**.**cno **=** 'CS114'

**and** s1**.**sno **not** **in** **(**

**select** s2**.**sno

**from** student s2**,** take t3**,** take t4**,** take t5

**where** s2**.**sno **=** t3**.**sno

**and** s2**.**sno **=** t4**.**sno

**and** s2**.**sno **=** t5**.**sno

**and** t3**.**cno **>** t4**.**cno

**and** t4**.**cno **>** t5**.**cno

**)**

Mine:

**select** s**.**sname**,** **count(**t**.**cno**)**

**from** students s**,** take t**,**

**(**

**select** sno**,** **count(**cno**)**

**from** take

**where** cno **IN** **(**'CS112'**,**'CS114'**)**

**group** **by** sno

**having** **count(**cno**)** **=** 2

**)** x

**where** s**.**sno **=** x**.**sno

**and** s**.**sno **=** t**.**sno

**group** **by** s**.**sname

**having** **count(**t**.**cno**)** **=** 2**;**

1. Anthony Molinario:

**select** s1**.\***

**from** student s1

**where** 2 **=** **(select** **count(\*)**

**from** student s2

**where** s2**.**age **<** s1**.**age**)**

David Rozenshtein:

**select** s5**.\***

**from** student s5**,**

student s6**,**

student s7

**where** s5**.**age **>** s6**.**age

**and** s6**.**age **>** s7**.**age

**and** s5**.**sno **not** **in** **(**

**select** s1**.**sno

**from** student s1**,**

student s2**,**

student s3**,**

student s4

**where** s1**.**age **>** s2**.**age

**and** s2**.**age **>** s3**.**age

**and** s3**.**age **>** s4**.**age

**)**

Mine:

**select** a**.**sname

**from** students a**,** students b

**where** a**.**sname **not** **in** **(**

**select** a**.**sname

**from** students a**,** students b

**where** a**.**age **-** b**.**age **>=** 3

**)**

**and** a**.**age **-** b**.**age **>=** 2**;**

1. Anthony Molinario:

**select** s**.**sno**,** s**.**sname**,** s**.**age

**from** student s**,** take t

**where** s**.**sno **=** t**.**sno

**group** **by** s**.**sno**,** s**.**sname**,** s**.**age

**having** **count(**t**.**cno**)** **=** **(select** **count(\*)**

**from** courses**)**

David Rozenshtein:

**select** **\***

**from** student

**where** sno **not** **in** **(**

**select** s**.**sno

**from** student s**,** courses **case**

**where** **(**s**.**sno**,** c**.**cno**)** **not** **in** **(select** sno**,** cno

**from** take**)**

**)**

Mine:

**select** sname**,** **count(**cno**)**

**from** students a**,** take b

**where** a**.**sno **=** b**.**sno

**group** **by** sname

**having** **count(**cno**)** **=** **(**

**select** **count(**cno**)**

**from** courses

**);**

1. Anthony Molinario:

**select** **\***

**from** student

**where** age **=** **(select** **max(**age**)**

**from** student**)**

David Rozenshtein:

**select** **\***

**from** student

**where** age **not** **in** **(select** a**.**age

**from** student a**,** student b

**where** a**.**age **<** b**.**age**)**

Mine:

**select** sname**,**

age

**from**

**(**

**select** **dense\_rank()**over**(order** **by** age **desc)** age\_rk**,**

sname**,**

age

**from** students

**)**

**where** age\_rk **=** 1**;**