Self-test SQL Workshop

Document: e0087test.fm

3 September 2019

ABIS Training & Consulting
Diestsevest 32 / 4b
B–3000 Leuven
Belgium

INTRODUCTION TO THE SELF-TEST SQL WORKSHOP

Instructions

The aim of this test is to check if you have a thorough knowledge of SQL. After the test you will know whether the 2-day SQL workshop is still worthwhile for you to follow, or whether you can immediately enrol for the 1-day SQL for advanced users.

This test consists of 15 questions (mainly multiple choice). Sometimes multiple answers must be given in which case this will clearly be indicated. Write down your responses and compare them with the solutions given on the last page. This test will take about half an hour.

Remarks:

- The SQL syntax used in the questions is ANSI/ISO SQL:2003, currently supported by all platforms (Db2, Oracle, MySQL, MariaDB, SQL Server, PostgreSQL, Hive, ...)
- This is an advanced SQL test. If you are unsure about your basic SQL skills, you should better start with the self-test for the "Fundamentals of SQL and relational databases" course.

Table and column information

The questions are based on the following tables:

- **COURSES**: describes all the courses that can be organised.
- **SESSIONS**: describes courses organised at a certain moment.
- **PERSONS**: describes all persons (instructors, enrollees, other persons).
- **ENROLMENTS**: contains all information on enrollees and their sessions.

The following relations exist between the tables:

- **COURSES - SESSIONS**
  - obtain course information for a certain session
  - COURSES.CID = SESSIONS.S_CID
- **SESSIONS - ENROLMENTS**
  - obtain session information for a certain enrolment
  - SESSIONS.SNO = ENROLMENTS.E_SNO
- **PERSONS - SESSIONS**
  - obtain person information for the instructor of a certain session
  - PERSONS.PNO = SESSIONS.SINS_PNO
- **PERSONS - ENROLMENTS**
  - obtain person information for a certain enrollee
  - PERSONS.PNO = ENROLMENTS.E_PNO

Table content and column descriptions

- **COURSES** table

CID	CTITLE	CDUR
7890	Db2	5
7910	Unix	4
8500	Oracle	5
8000	SQLServer	5
9000	SQL workshop	3

  - **CID**: required, alphanumeric: course number (primary key)
  - **CTITLE**: required, alphanumeric: course title
  - **CDUR**: required, numeric: course duration (in days).

- **SESSIONS** table (8 rows)

SNO	S_CID	SDATE	SINS_PNO	SCANCEL
10	7890	2015-12-02	3	(NULL)
11	7910	2015-11-04	1	(NULL)
12	7890	.2016-01-08	3	C
13	7890	2016-02-02	3	(NULL)
14	8000	2016-04-05	2	C
15	7910	2016-01-08	36	C
16	8500	2016-04-05	36	(NULL)
17	9000	2016-06-07	36	(NULL)

  - **SNO**: required, numeric: session number (primary key).
  - **S_CID**: optional, alphanumeric: course number (foreign key to COURSES).
  - **SDATE**: optional: start date of the session.
  - **SINS_PNO**: required, numeric: session instructor (foreign key to PERSONS).
  - **SCANCEL**: optional: indicates if the session is cancelled (“C” means cancelled, empty (NULL) means not cancelled).

- **PERSONS** table (19 rows)

PNO	PNAME	P_CONO
1	SMITHS	3
2	TAVERNIER	3
3	DE KEYSER	3
4	HEBBELYNCK	5
5	VAN DE BROECK	5
6	VAN HEIJKOOP	10
7	DE WINDT	2
8	SPENSER	10
9	BENOIT	1
10	BENOIT	1
11	LOOSE	(NULL)
13	PARKER	6
15	DEHEM	7
17	PIELAGE	4
18	GELADE	2
33	BUENK	9
36	ADAMSON	8
45	MOORS	4
50	MAK	(NULL)

  - **PNO**: required, numeric: person number (primary key).
  - **PNAME**: optional, alphanumeric: name.
  - **P_CONO**: optional, numeric: number of the company the person works for.

- **ENROLMENTS** table (14 rows, 9 different enrollees)

E_SNO	E_PNO	ECANCEL
10	4	(NULL)
10	7	C
11	45	(NULL)
11	13	(NULL)
12	4	(NULL)
13	15	C
13	36	(NULL)
14	3	(NULL)
14	18	C
14	1	(NULL)
15	4	(NULL)
15	7	(NULL)
16	3	(NULL)
16	18	(NULL)

  - **E_SNO**: required, numeric: session number for the enrolment (foreign key to SESSIONS) (primary key together with E_PNO)
  - **E_PNO**: required, numeric: the enrollee (foreign key to PERSONS) (primary key together with E_SNO)
  - **ECANCEL**: optional: “C” when enrolment was cancelled, NULL if not cancelled.

QUESTIONS SELF-TEST SQL WORKSHOP

1. Which ones of the following queries produce exactly 1 result row? *[2 correct answers.]*

[_] [a]

```
SELECT COUNT(*)
FROM   PERSONS
WHERE  PNO > 100
```

[_] [b]

```
SELECT PNO, COUNT(*)
FROM   PERSONS
WHERE  PNO = 2
```

[_] [c]

```
SELECT COUNT(*)
FROM   PERSONS
GROUP BY PNO
```

[_] [d]

```
SELECT PNAME
FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
WHERE  PNO = 36
```

[_] [e]

```
SELECT PNAME
FROM   PERSONS LEFT OUTER JOIN ENROLMENTS ON PNO = E_PNO
WHERE  PNO = 2
GROUP BY PNAME
```

[_] [f]

```
SELECT SUM(CDUR)
FROM   COURSES, SESSIONS, ENROLMENTS
WHERE  CID = S_CID AND SNO = E_SNO
GROUP BY CID
```

2. How many result rows are produced by this query?

```
SELECT E_SNO
FROM   ENROLMENTS
UNION
SELECT SNO
FROM   SESSIONS
WHERE  SNO BETWEEN 15 AND 17
```

Answer: .....

3. Which queries produce the following table as the result? *[3 correct answers.]*

PNO	PNAME	
1	SMITHS	ENROLLEE
3	DE KEYSER	ENROLLEE
4	HEBBELYNCK	ENROLLEE
7	DE WINDT	ENROLLEE
13	PARKER	ENROLLEE
15	DEHEM	ENROLLEE
18	GELADE	ENROLLEE
36	ADAMSON	ENROLLEE
45	MOORS	ENROLLEE
1	SMITHS	INSTRUCTOR
2	TAVERNIER	INSTRUCTOR
3	DE KEYSER	INSTRUCTOR
36	ADAMSON	INSTRUCTOR

[_] [a]

```
SELECT PNO, PNAME, 'ENROLLEE OR INSTRUCTOR'
FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
               INNER JOIN ENROLMENTS ON PNO = E_PNO
ORDER BY 3, 1
```

[_] [b]

```
SELECT PNO, PNAME, CASE PNO WHEN E_PNO THEN 'ENROLLEE' ELSE 'INSTRUCTOR' END
FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
               INNER JOIN ENROLMENTS ON PNO = E_PNO
ORDER BY 3, 1
```

[_] [c]

```
 SELECT PNO, PNAME, 'INSTRUCTOR'
 FROM   PERSONS
 WHERE  PNO IN (SELECT SINS_PNO
                FROM   SESSIONS)
UNION ALL
 SELECT PNO, PNAME, 'ENROLLEE'
 FROM   PERSONS INNER JOIN ENROLMENTS ON PNO = E_PNO
ORDER BY 3, 1
```

[_] [d]

```
 SELECT DISTINCT PNO, PNAME, 'INSTRUCTOR'
 FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
UNION ALL
 SELECT PNO, PNAME, 'ENROLLEE'
 FROM   PERSONS
 WHERE  PNO IN (SELECT E_PNO
                FROM   ENROLMENTS)
ORDER BY 3, 1
```

[_] [e]

```
 SELECT PNO, PNAME, 'INSTRUCTOR'
```

```
 FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
UNION
 SELECT PNO, PNAME, 'ENROLLEE'
 FROM   PERSONS
 WHERE  PNO IN (SELECT E_PNO
                FROM   ENROLMENTS)
ORDER BY 3, 1
```

[_] [f]

```
 SELECT DISTINCT PNO, PNAME, 'INSTRUCTOR'
 FROM   PERSONS INNER JOIN SESSIONS ON PNO = SINS_PNO
UNION
 SELECT PNO, PNAME, 'ENROLLEE'
 FROM   PERSONS P
 WHERE  EXISTS (SELECT E_PNO
                FROM   ENROLMENTS
                WHERE  E_PNO = P.PNO)
ORDER BY 3, 1
```

4. How many result rows are produced by the following query?

```
SELECT DISTINCT PNO
FROM   PERSONS LEFT OUTER JOIN ENROLMENTS ON PNO = E_PNO
```

Answer: .....

5. Which queries give an answer to the following question? *[3 correct answers.]*

Give the number of all sessions for which none of the enrolments have been cancelled.

[_] [a]

```
SELECT DISTINCT SNO
FROM   SESSIONS, ENROLMENTS
WHERE  SNO = E_SNO AND ECANCEL IS NULL
```

[_] [b]

```
SELECT DISTINCT SNO
FROM   SESSIONS, ENROLMENTS
WHERE  SNO = E_SNO AND ECANCEL IS NOT NULL
```

[_] [c]

```
WITH E AS (SELECT E_SNO
           FROM   ENROLMENTS
           WHERE  ECANCEL IS NOT NULL)
SELECT SNO
FROM   SESSIONS LEFT OUTER JOIN E ON SNO = E_SNO
WHERE  E_SNO IS NULL
```

[_] [d]

```
SELECT SNO
FROM   SESSIONS
WHERE  SNO IN (SELECT E_SNO
               FROM   ENROLMENTS
               WHERE  ECANCEL IS NULL)
```

[_] [e]

```
SELECT SNO FROM SESSIONS
EXCEPT                  -- or MINUS when using Oracle
SELECT E_SNO FROM ENROLMENTS WHERE ECANCEL IS NOT NULL
```

[_] [f]

```
SELECT SNO
FROM   SESSIONS S
WHERE  NOT EXISTS (SELECT 1
                   FROM   ENROLMENTS
                   WHERE  E_SNO = S.SNO AND ECANCEL IS NOT NULL)
```

[_] [g]

```
SELECT SNO
FROM   SESSIONS INNER JOIN ENROLMENTS ON SNO = E_SNO
WHERE  ECANCEL IS NULL
```

[_] [h]

```
SELECT SNO
FROM   SESSIONS INNER JOIN ENROLMENTS ON SNO = E_SNO
WHERE  ECANCEL IS NOT NULL
```

6. Which queries produce the following table “all enrollees”? *[3 correct answers.]*

PNAME
SMITHS
DE KEYSER
HEBBELYNCK
ADAMSON
DE WINDT
PARKER
DEHEM
GELADE
MOORS

[_] [a]

```
SELECT PNAME  FROM PERSONS
WHERE  PNO IN (SELECT E_PNO AS PNO  FROM ENROLMENTS)
```

[_] [b]

```
SELECT PNAME
FROM   PERSONS INNER JOIN ENROLMENTS ON PNO = E_PNO
```

[_] [c]

```
SELECT PNAME  FROM PERSONS
WHERE  PNO = ANY (SELECT E_PNO FROM ENROLMENTS)
```

[_] [d]

```
SELECT PNAME  FROM PERSONS
WHERE  EXISTS (SELECT E_PNO FROM ENROLMENTS)
```

[_] [e]

```
SELECT PNAME
FROM   (SELECT E_PNO FROM ENROLMENTS WHERE  E_PNO IS NOT NULL) E
       INNER JOIN PERSONS  ON PNO = E.E_PNO
```

[_] [f]

```
SELECT PNAME
FROM   PERSONS LEFT OUTER JOIN ENROLMENTS ON PNO = E_PNO
GROUP BY PNAME
```

[_] [g]

```
SELECT PNAME
FROM   PERSONS RIGHT OUTER JOIN ENROLMENTS ON PNO = E_PNO
GROUP BY PNAME
```

7. Which question corresponds best to the following query?

```
SELECT P_CONO, COUNT(*)
FROM   PERSONS P
WHERE  EXISTS (SELECT SNO
               FROM   SESSIONS
               WHERE  SINS_PNO = P.PNO)
GROUP BY P_CONO
```

O (a) Give per instructor the number of sessions he teaches. Give also the company where he is employed.

O (b) Give per company the number of employees who followed a course.

O (c) Give the number of sessions per course, and also the company where the instructor is employed.

O (d) Give the number of instructors per company.

8. Which ones of the following queries are equivalent to this query? *[2 correct answers.]*

```
SELECT PNAME
FROM   PERSONS
WHERE  PNO = ( SELECT MAX(PNO)  FROM PERSONS )
```

[_] [a]

```
SELECT PNAME
FROM   PERSONS
WHERE  PNO >=  ANY (SELECT PNO  FROM PERSONS)
```

[_] [b]

```
SELECT PNAME
FROM   PERSONS
WHERE  PNO >= ALL (SELECT PNO  FROM PERSONS)
```

[_] [c]

```
SELECT PNAME
FROM   PERSONS P1
WHERE  EXISTS (SELECT MAX(PNO)
               FROM   PERSONS P2
               WHERE  P1.PNO = P2.PNO)
```

[_] [d]

```
SELECT PNAME, MAX(PNO)
FROM   PERSONS
GROUP BY PNAME
```

[_] [e]

```
SELECT P1.PNAME
FROM   PERSONS P1
       LEFT OUTER JOIN
       PERSONS P2      ON P1.PNO < P2.PNO
GROUP BY P1.PNO, P1.PNAME
HAVING COUNT(P2.PNO) = 0
```

9. Which queries give an answer to the following question? *[3 correct answers.]*

Give the list of all courses, also those for which no session has been planned yet.
Give also all corresponding session numbers and the date on which each session starts.

[_] [a]

```
SELECT CID, SNO, SDATE
FROM   COURSES INNER JOIN SESSIONS ON CID = S_CID
```

[_] [b]

```
SELECT S_CID, SNO, SDATE
FROM   SESSIONS
```

[_] [c]

```
SELECT CID, SNO, SDATE
FROM   COURSES LEFT OUTER JOIN SESSIONS ON CID = S_CID
```

[_] [d]

```
SELECT CID, SNO, SDATE
FROM   COURSES RIGHT OUTER JOIN SESSIONS ON CID = S_CID
```

[_] [e]

```
WITH S AS (SELECT SNO, S_CID, SDATE
           FROM   SESSIONS
           WHERE  S_CID IS NOT NULL)
 SELECT CID, SNO, SDATE
 FROM   COURSES INNER JOIN S ON CID = S_CID
UNION ALL
 SELECT CID, 0, CAST(NULL AS DATE)
 FROM   COURSES
 WHERE  CID NOT IN (SELECT S_CID FROM S)
```

[_] [f]

```
 SELECT C.CID, S.SNO, S.SDATE
 FROM   (SELECT CID FROM COURSES) C
        LEFT OUTER JOIN
        (SELECT SNO, S_CID, SDATE FROM SESSIONS) S
        ON  S_CID = CID
```

[_] [g]

```
 SELECT CID, SNO, SDATE
 FROM   COURSES INNER JOIN SESSIONS ON CID = S_CID
UNION ALL
 SELECT S_CID, SNO, SDATE
 FROM   SESSIONS
 WHERE  S_CID IS NULL
```

10. Which query implements the following question?

Give the names of all instructors who have in addition also followed more than 1 course.

O (a)

```
SELECT PNAME  FROM  PERSONS
WHERE  PNO IN (SELECT E_PNO
               FROM   ENROLMENTS INNER JOIN SESSIONS ON E_SNO = SNO
               WHERE  E_PNO = SINS_PNO
                 AND  ECANCEL IS NULL
                 AND  SCANCEL IS NULL
               GROUP BY E_PNO  HAVING COUNT(*) > 1)
```

O (b)

```
SELECT PNAME  FROM PERSONS
WHERE  PNO IN (SELECT SINS_PNO
               FROM   SESSIONS
               WHERE  SCANCEL IS NULL
                 AND  SNO IN (SELECT E_SNO
                              FROM   ENROLMENTS
                              WHERE  ECANCEL IS NULL
                              GROUP BY E_PNO  HAVING COUNT(*) > 1))
```

O (c)

```
SELECT PNAME
FROM   PERSONS INNER JOIN
       (SELECT E_PNO FROM ENROLMENTS
         WHERE ECANCEL IS NULL
           AND E_SNO IN (SELECT SNO FROM SESSIONS  WHERE SCANCEL IS NULL)
        GROUP BY E_PNO  HAVING COUNT(*) > 1)   E
       ON E_PNO = PNO
WHERE  PNO IN (SELECT SINS_PNO FROM SESSIONS)
```

O (d)

```
SELECT PNAME
FROM   PERSONS INNER JOIN ENROLMENTS  ON PNO = E_PNO
               INNER JOIN SESSIONS S1 ON E_SNO = S1.SNO
               INNER JOIN SESSIONS S2 ON PNO = S1.SINS_PNO
WHERE  S1.SCANCEL IS NULL  AND  ECANCEL IS NULL
GROUP BY E_PNO, PNAME  HAVING COUNT(*) > 1
```

O (e)

```
WITH P AS (SELECT PNO, PNAME FROM PERSONS),
     E AS (SELECT E_PNO, E_SNO FROM ENROLMENTS WHERE ECANCEL IS NULL),
     S AS (SELECT SNO, SINS_PNO FROM SESSIONS  WHERE SCANCEL IS NULL)
SELECT PNAME
FROM   P INNER JOIN E    ON PNO = E_PNO
         INNER JOIN S S1 ON E_SNO = S1.SNO
         INNER JOIN S S2 ON PNO = S2.SINS_PNO
GROUP BY E_PNO, PNAME  HAVING COUNT(*) > 1
```

11. What can be said about this query?

```
 SELECT SNO, PNAME, SDATE
 FROM   SESSIONS, PERSONS
 WHERE  SINS_PNO = PNO
UNION
 SELECT E_PNO, PNAME
 FROM   PERSONS, ENROLMENTS
 WHERE  PNO = E_PNO
ORDER BY 1
```

O (a) Query cannot be executed (gives a syntax error).

O (b) Query can be executed and makes sense (according to the table and column definitions).

O (c) Query can be executed but returns nonsense.

12. What can be said about this query?

```
SELECT SNO, SDATE, PNAME
FROM   SESSIONS INNER JOIN ENROLMENTS ON SNO = E_SNO
                INNER JOIN PERSONS ON P_CONO = E_PNO
WHERE  ECANCEL IS NULL
  AND  SCANCEL IS NULL
```

O (a) Query cannot be executed (gives a syntax error).

O (b) Query can be executed and makes sense (according to the table and column definitions).

O (c) Query can be executed but returns nonsense.

13. What can be said about this query?

```
SELECT SNO, SDATE, S_CID
FROM   SESSIONS S
WHERE  SCANCEL IS NOT NULL
  AND  SDATE = (SELECT MAX(SDATE)
                FROM   SESSIONS
                WHERE  S_CID = S.S_CID)
```

O (a) Query cannot be executed (gives a syntax error).

O (b) Query can be executed and makes sense (according to the table and column definitions).

O (c) Query can be executed but returns nonsense.

14. What can be said about this query?

```
SELECT PNAME, COUNT(*)
FROM   PERSONS INNER JOIN SESSIONS ON SINS_PNO = PNO
```

O (a) Query cannot be executed (gives a syntax error).

O (b) Query can be executed and makes sense (according to the table and column definitions).

O (c) Query can be executed but returns nonsense.

15. What can be said about this query?

```
SELECT (SELECT COUNT(*) AS nr_8000
        FROM   SESSIONS
        WHERE  S_CID = '8000')
       * 100.0 / COUNT(S_CID) AS percent_8000
FROM   SESSIONS
```

O (a) Query cannot be executed (gives a syntax error).

O (b) Query can be executed and makes sense (according to the table and column definitions).

O (c) Query can be executed but returns nonsense.

EVALUATION.

Here are the correct answers to all questions:

1. a e
2. 8
3. d e f
4. 19
5. c e f
6. a c g
7. d
8. b e
9. c e f
10. c
11. a
12. c
13. b
14. a
15. b

Give yourself 1 point for each correctly answered question; for multiple answer questions, all answers should be correct.

When your score is 80% or above, you are ready for our Advanced SQL course.

When your score is between 50% and 80%, following the course SQL workshop will allow you to refine your SQL knowledge.

When your score is less than 50%, following the course SQL workshop is advisable. You will get a high return from this course. Be sure your basic SQL and RDBMS knowledge is sufficient: fill out the corresponding self-test to verify this.