

CSCI 3410 – Last Commands

Clément Aubert

February 7, 2018

```
1
2  /*
3   * You need the CREATE and UPDATE statements from
4   * http://spots.augusta.edu/caubert/teaching/2018/spring/csci3410/snippets/2018_02_05/
5   * to be able to use this code.
6   */
7
8  USE HW_UNIV;
9
10 CREATE TABLE GRADE(
11     Login VARCHAR(25),
12     Grade DECIMAL(3, 2), -- 3 digits (the precision), 2 decimals (the scale), i.e. from
13     ↪ -9.99 to 9.99
14     FOREIGN KEY (Login) REFERENCES STUDENT(Login)
15 );
16
17 INSERT INTO GRADE VALUES
18     ('jrakesh', 3.85),
19     ('svlatka', NULL),
20     ('cjoella', 4.0),
21     ('aalyx', 3.3);
22
23 INSERT INTO GRADE VALUES
24     ('jrakesh', 2.85),
25     ('jrakesh', 2.85)
26 ;
27
28 /*
29 SELECT * FROM STUDENT;
30
31 +-----+-----+-----+-----+
32 | Login   | Name           | Registered | Major |
33 +-----+-----+-----+-----+
34 | aalyx   | Ava Alyx       | 2012-10-11 | CYBR  |
35 | cjoella | Candice Joella | 2012-02-12 | CYBR  |
36 | jrakesh | Jalal Rakesh   | 2017-12-01 | CS    |
37 | svlatka | Sacnite Vlatka | 2015-03-12 | MATH  |
38 +-----+-----+-----+-----+
```

```

37 SELECT * FROM PROF;
38 +-----+-----+-----+
39 | Login   | Name                | Department |
40 +-----+-----+-----+
41 | aturing  | Alan Turing         | CS        |
42 | bgates   | Bill Gates          | CYBR      |
43 | caubert  | Clément Aubert     | CS        |
44 | perdos   | Paul Erdős          | MATH      |
45 +-----+-----+-----+
46 SELECT * FROM DEPARTMENT;
47 +-----+-----+-----+
48 | Code | Name                | Head      |
49 +-----+-----+-----+
50 | CS   | Computer Science   | bgates    |
51 | CYBR | Cyber Security     | bgates    |
52 | MATH | Mathematics        | aturing   |
53 +-----+-----+-----+
54 SELECT * FROM GRADE;
55 +-----+-----+
56 | Login   | Grade |
57 +-----+-----+
58 | jrakesh | 3.85  |
59 | svlatka | NULL  |
60 | cjoella | 4.00  |
61 | aalyx   | 3.30  |
62 | jrakesh | 2.85  |
63 | jrakesh | 2.85  |
64 +-----+-----+
65 */
66
67 -- ALL / DISTINCT
68 SELECT DISTINCT Major FROM STUDENT;
69
70 -- Union
71 (SELECT Login FROM STUDENT) UNION (SELECT Login FROM PROF);
72
73 -- ORDER BY
74 SELECT Login FROM GRADE WHERE Grade > 3.0 ORDER BY Grade;
75
76 SELECT Login FROM GRADE WHERE Grade > 3.0 ORDER BY Grade DESC;
77
78 SELECT Login, Major FROM STUDENT ORDER BY Major, Name;
79
80 -- Aggregate functions
81 SELECT MAX(Registered) FROM STUDENT;
82
83 SELECT COUNT(Name) FROM STUDENT;
84

```

```
85  -- Aliases for columns
86  SELECT Login AS Username FROM PROF;
87
88  --Three-valued logic
89  SELECT Login FROM GRADE WHERE Grade IS NULL;
90
91  -- Select-project-join
92  SELECT Login FROM PROF, DEPARTMENT WHERE DEPARTMENT.Name = 'Mathematics' AND
    ↪ PROF.Department = DEPARTMENT.Code;
93
94  SELECT Name FROM STUDENT, GRADE WHERE Grade > 3.0 AND STUDENT.Login = GRADE.Login;
95
96  SELECT PROF.Name FROM PROF, DEPARTMENT, STUDENT WHERE STUDENT.Name = 'Ava Alyx' AND
    ↪ STUDENT.Major = DEPARTMENT.Code AND DEPARTMENT.Head = PROF.Login;
97
98  -- Aliases or tuple variables
99  SELECT A.Name FROM PROF AS A, DEPARTMENT, STUDENT AS B WHERE B.Name = 'Ava Alyx' AND
    ↪ B.Major = DEPARTMENT.Code AND DEPARTMENT.Head = A.Login;
100
101  SELECT Others.Login FROM GRADE AS Mine, GRADE as Others WHERE Mine.Login = 'aalyx' and
    ↪ Mine.Grade < Others.Grade;
102
103  SELECT Fellow.Name AS 'Fellow of Ava' FROM STUDENT AS Me, STUDENT AS Fellow WHERE Me.Name =
    ↪ 'Ava Alyx' AND Fellow.Major = Me.Major AND NOT Fellow.Name = 'Ava Alyx';
104
105  -- Nested Queries
106
107  SELECT Login FROM GRADE WHERE Grade >
108      (SELECT AVG(Grade) FROM GRADE);
109
110  SELECT Login FROM GRADE WHERE Grade >= ALL
111      (SELECT Grade FROM GRADE WHERE Grade IS NOT NULL);
112
113  SELECT Login FROM PROF WHERE DEPARTMENT IN
114      (SELECT Major FROM STUDENT WHERE Login LIKE '%a');
```