CSCI 3410 – Last Commands

Clément Aubert

February 7, 2018

/*
* You need the CREATE and UPDATE statements from
* http://spots.augusta.edu/caubert/teaching/2018/spring/csci3410/snippets/2018_02_05/
* to be able to use this code.
*/

USE HW_UNIV;

CREATE TABLE GRADE(
    Login VARCHAR(25),
    Grade DECIMAL(3, 2), -- 3 digits (the precision), 2 decimals (the scale), i.e. from -9.99 to 9.99
    FOREIGN KEY (Login) REFERENCES STUDENT(Login)
);

INSERT INTO GRADE VALUES
    ('jrakesh', 3.85),
    ('svlatka', NULL),
    ('cjoella', 4.0),
    ('aalyx', 3.3);

INSERT INTO GRADE VALUES
    ('jrakesh', 2.85),
    ('jrakesh', 2.85)
;

/*
SELECT * FROM STUDENT;

<table>
<thead>
<tr>
<th>Login</th>
<th>Name</th>
<th>Registered</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>aalyx</td>
<td>Ava Alyx</td>
<td>2012-10-11</td>
<td>CYBR</td>
</tr>
<tr>
<td>cjoella</td>
<td>Candice Joella</td>
<td>2012-02-12</td>
<td>CYBR</td>
</tr>
<tr>
<td>jrakesh</td>
<td>Jalal Rakesh</td>
<td>2017-12-01</td>
<td>CS</td>
</tr>
<tr>
<td>svlatka</td>
<td>Sacnite Vlatka</td>
<td>2015-03-12</td>
<td>MATH</td>
</tr>
</tbody>
</table>

*/
```sql
SELECT * FROM PROF;
+---------+-----------------+------------+
| Login   | Name            | Department |
+---------+-----------------+------------+
| aturing | Alan Turing     | CS         |
| bgates  | Bill Gates      | CYBR       |
| caubert | Clément Aubert  | CS         |
| perdos  | Paul Erdős      | MATH       |
+---------+-----------------+------------+
SELECT * FROM DEPARTMENT;
+------+------------------+---------+
| Code | Name              | Head    |
+------+------------------+---------+
| CS   | Computer Science  | bgates  |
| CYBR | Cyber Security    | bgates  |
| MATH | Mathematics       | aturing |
+------+------------------+---------+
SELECT * FROM GRADE;
+---------+-------+
| Login   | Grade |
+---------+-------+
| jrakesh | 3.85  |
| svlatka | NULL  |
| cjoella | 4.00  |
| aalyx   | 3.30  |
| jrakesh | 2.85  |
| jrakesh | 2.85  |
+---------+-------+
*/
-- ALL / DISTINCT
SELECT DISTINCT Major FROM STUDENT;
-- Union
(SELECT Login FROM STUDENT) UNION (SELECT Login FROM PROF);
-- ORDER BY
SELECT Login FROM GRADE WHERE Grade > 3.0 ORDER BY Grade;
SELECT Login FROM GRADE WHERE Grade > 3.0 ORDER BY Grade DESC;
SELECT Login, Major FROM STUDENT ORDER BY Major, Name;
-- Aggregate functions
SELECT MAX(Registered) FROM STUDENT;
SELECT COUNT(Name) FROM STUDENT;
```
-- Aliases for columns
SELECT Login AS Username FROM PROF;

-- Three-valued logic
SELECT Login FROM GRADE WHERE Grade IS NULL;

-- Select-project-join
SELECT Login FROM PROF, DEPARTMENT WHERE DEPARTMENT.Name = 'Mathematics' AND
  PROF.Department = DEPARTMENT.Code;

SELECT Name FROM STUDENT, GRADE WHERE Grade > 3.0 AND STUDENT.Login = GRADE.Login;

SELECT PROF.Name FROM PROF, DEPARTMENT, STUDENT WHERE STUDENT.Name = 'Ava Alyx' AND
  STUDENT.Major = DEPARTMENT.Code AND DEPARTMENT.Head = PROF.Login;

-- Aliases or tuple variables
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;

SELECT Others.Login FROM GRADE AS Mine, GRADE AS Others WHERE Mine.Login = 'aalyx' and
  Mine.Grade < Others.Grade;

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';

-- Nested Queries
SELECT Login FROM GRADE WHERE Grade >
  (SELECT AVG(Grade) FROM GRADE);

SELECT Login FROM GRADE WHERE Grade >= ALL
  (SELECT Grade FROM GRADE WHERE Grade IS NOT NULL);

SELECT Login FROM PROF WHERE DEPARTMENT IN
  (SELECT Major FROM STUDENT WHERE Login LIKE '%$a$');