Lab and Central Exercises

Relational Query Language

Chapter 7, Part 3
Subqueries, Window Functions and Common Table Expressions

1. University: Which professors do not teach? Please, write 3 syntactically different but semantically equivalent options, using join, correlated subquery and uncorrelated subquery.

knirsch@kiu.edu.ge

2. University: Display the teaching load (that is the contact hours) that each professor has. Also display each professor's percentage of the total teaching load (total contact hours).

Write one query as subquery and another query with window function.

Add the proferssor name to one of your your output tables.

knirsch@kiu.edu.ge

3. University: Does the university have students that are enrolled in all courses?

3.1 Create a table enrollment_test that you can verify your query with.

Use command:

create table <newTableName> like <oldTableName>;

3.2 Populate your new table with the cross join result from studID and courseID (See slide 6 of SQL Part 2 for exact command)

3.3 Delete a couple of rows so that not all students take all courses.

3.4 Write the query – either with CTE or subquery. (There are different options.)

knirsch@kiu.edu.ge

4. Recursive CTE: Implement and run the example of lecture: Follow the stream MtisChala and display the river it flows into and the river that the intake river flows into and so on ....

knirsch@kiu.edu.ge

Subquery / CTE

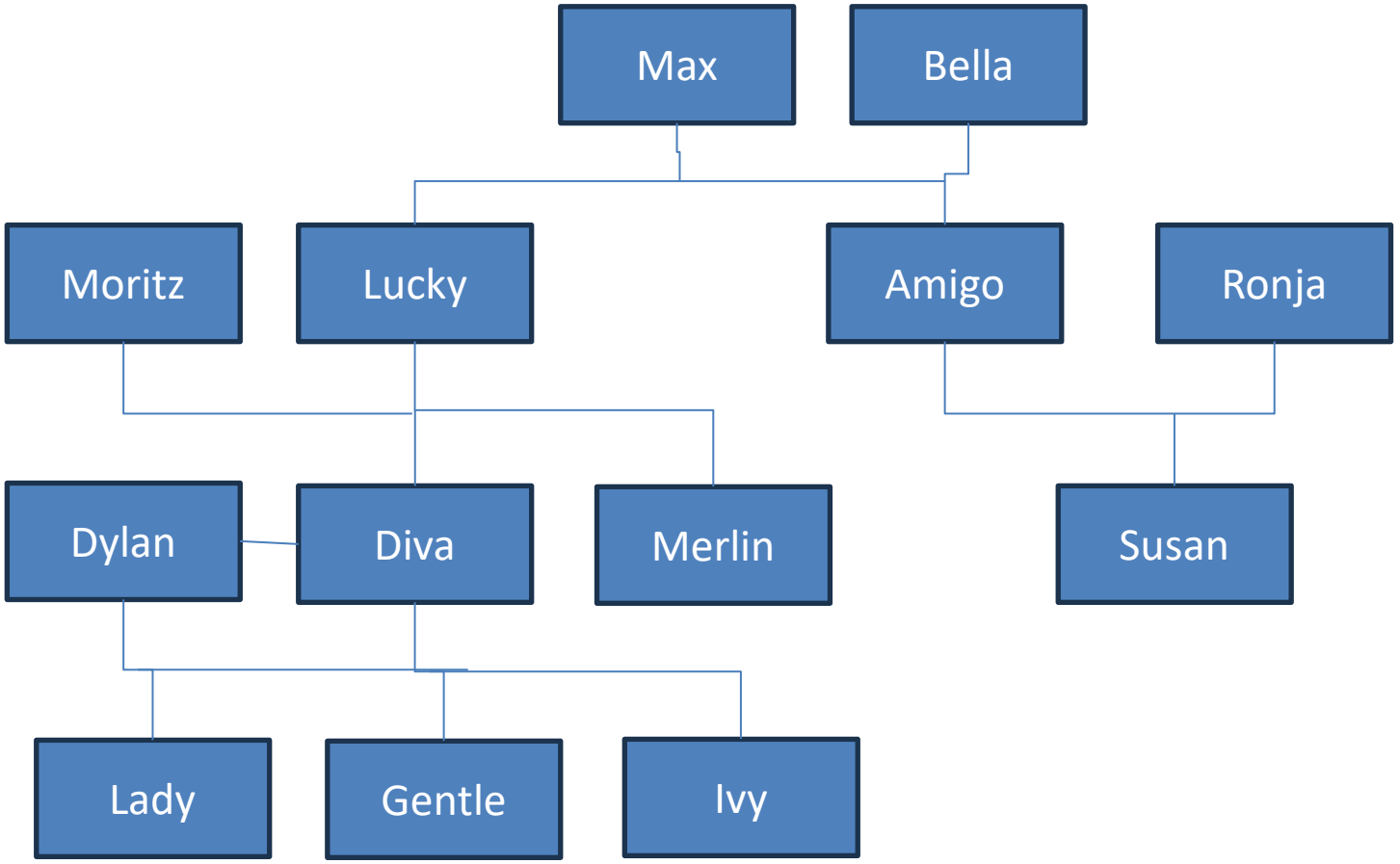
5. University Database:

Which courses does a student need to pass before the student is allowed to take course 5052? Show courseID and courseName of all **required** courses.

Central Exercises
Relational Query Language

Chapter 7, Part 3
Subqueries, Window Functions and Common Table Expressions

6. Horse Family Tree



6. Horse Family Tree

6.1: Implement table horse in some database

```
CREATE TABLE `horse` (
  `horseName` varchar(50) NOT NULL,
  `DoB` date DEFAULT NULL,
  `gender` enum('f','m') DEFAULT NULL,
  `father` varchar(50) DEFAULT NULL,
  `mother` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`horseName`),
  KEY `Index 2` (`father`),
  KEY `Index 3` (`mother`),
  CONSTRAINT `fatherIndex` FOREIGN KEY (`father`) REFERENCES `horse` (`horseName`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `motherIndex` FOREIGN KEY (`mother`) REFERENCES `horse`
(`horseName`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8mb3_general_ci
```


knirsch@kiu.edu.ge

6. Horse Family Tree

6.2 Populate as given on slide

(For the lazy ones, the table and data is also on TEAMS.)

6.3 Follow the family tree of Gentle – fatherline.
(You get 2 result rows.)

6.4 Do the same for Susan – fatherline
(3 result rows)

6.5 Follow the family tree of Gentle – motherline
(4 result rows)

6.6 Follow the family tree of Gentle – father and mother
(You need 2 recursive parts that are also appended together with another union)

knirsch@kiu.edu.ge

Subquery / CTE

7. University Database:

7.1 Add a column "points" to the examination table
(Attention: Do not name the column "point" as this is a keyword.)

7.2 Insert a couple of rows into examination: for course 5001 and fill in points.

7.3. Which students scored less than average points in course 5001 (Foundation)?
Display studentID, student name, semester and grade and points

Write the query once with subquery and once with CTE.

Subquery / CTE

8. University

Which students are enrolled in all 4-contact hour courses?
(Make sure you have appropriate test data!)