| **#** | **Question** | **Table(s)** | **Marks** |
| --- | --- | --- | --- |
| 1. | List the required course prerequisites for COMP2280. Include the course codes and course names for each prerequisite in the result. Order the result by the prerequisite course code. | Course  CoursePrerequisiteAnd | 4 |
| 2. | List the employees whose last name begins with the letter “W”, using a LIKE predicate. Order the result by employee number. Display the employee’s number, first name, last name, schoolCode, and phone extension. | Employee,  Person | 5 |
| 3. | List the persons from the Person table who are not students. Show the person’s number, first name, last name, and city. Use LEFT OUTER JOIN. Order the results by the person’s last name. | Person,  Student | 5 |
| 4. | List the lab software installed on machines in room 2A205. List each software package’s unique identifier along with the software product’s name. Order the result by software product name. | LabSoftware  Software  Room | 5 |
| 5. | Assume we want to show the names of employees (faculty) who taught courses in a particular session. Show **CourseOffering** *sessionCode*, **CourseOffering** *courseNumber*, **Person** *firstName*, **Person** *lastName,* and Employee *extension* for courses offered in the Fall 2010 session (‘F10’ in the CourseOffering table). Order the results by *courseNumber*.  F10 CDEV1020 KATHY HUHN  F10 INFO1030 DALE KARSH  F10 INFO1570 RICK WHETSTONE  F10 INFO8000 ABDOLSAMAD KATHIRESAN  F10 PROG8080 LYNDOL YUROVIC  F10 PROG8090 RAKESH WEKENMANN  (6 row(s) affected) | CourseOffering  Person  Employee | 7 |
| 6. | List the names and email addresses of the students who have committed any academic offense, along with the reason for the offence (the Notes column in the StudentOffense table). Order the result by student number. | Student  StudentOffence  Person | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | Show **Employee** *e1.number* aliased as ‘Employee’ and **Employee** *e1.reportsTo* aliased as ‘Employee Reports To’ and **Employee** *e2.reportsTo* aliased as ‘Supervisor Reports To’. Order the results by *e1.number*. Do left outer self-join (*e1.reportsTo* = *e2.number*).  Employee Employee Reports To Supervisor Reports To  -------- ------------------- ---------------------  2117745 4427732 NULL  3214350 4427732 NULL  3455730 4427732 NULL  3514239 4427732 NULL  3566163 4427732 NULL  4427732 NULL NULL  5512736 6860242 NULL  6434468 NULL NULL  6860242 NULL NULL  6886352 8436780 NULL  7572373 4427732 NULL  8135170 4427732 NULL  8383259 6434468 NULL  8436780 NULL NULL  (14 row(s) affected | Employee | 7 |
| 8. | List user id’s for all employees whose last name starts with ‘K’. A user id consists of the first letter of the first name followed by the first seven letters of the last name, all in lower case. Alias the user id ‘User Id’, and sort the results in ascending ‘User ID’ order. Use either LEFT() or SUBSTRING() as you wish. Include the employee’s schoolCode and extension in the result. | Person  Employee | 5 |
| 9. | List the courses containing the word “Programming” in their title that are not currently offered. Order the result by course code. Do not use a LIKE predicate. Use a LEFT OUTER JOIN.  number hours credits name  PROG1780 90 6 Programming: Fundamentals  PROG2070 45 3 Programming: Software Quality Assurance  PROG2240 45 3 Programming: Java Web Technologies  PROG2250 60 4 Programming: Object Oriented Game Programming  PROG3050 75 5 Programming: Microsoft Enterprise Applications  PROG3060 45 3 Programming: Java Enterprise Applications  PROG3170 45 3 Programming: Distributed Applications Development  PROG3180 45 3 Programming: Mobile Applications | Course  CourseOffering | 7 |
|  | | | 50 |