W228/206

**DUBLIN INSTITUTE OF TECHNOLOGY**

**KEVIN STREET, DUBLIN 8**

**WINTER EXAMINATIONS 2012**

DATABASES 1

Ms. P. O’Byrne

Dr. D. Lillis

Mr. Brian Boyle

Friday 13 Jan 2012 1.00-3.00

Answer Question 1 (40%) and two others (30% each).

Read case study 1 on Page 2 before attempting questions 1.

Read the case study 2 on Page 2 before attempting questions 2 through 4.

There is a syntax table on the last page to assist you.

1. (a) List the primary entities for the Pine Valley DVD Club. (5 marks)



Case Study Pine Valley DVD club

(b) For each entity, state which attribute(s) you would use as a primary key. (5 marks)

(c) Show how the entities should be related to each other; by drawing an Entity Relationship Diagram (entities only), using IE notation. (5 marks)

(d) List the attributes for each of your entities, noting constraints. (5 marks)

(e) Draw a full ERD, complete with attributes, underlining primary keys and marking foreign keys with an asterisk. (10 marks)

(f) Assuming that you have created tables in your schema that correspond to your ERD, write a set of SQL statements to add the following information:

The member Kiera Knightley, (member’s name) of 12 Notting Hill, London, (member’s address) who joined the club on 1st January /2006, has rented a DVD of the movie entitled ‘Notting Hill’ (censor rating 12A) on 10/8/2011 for €3. It is due back on 14/8/2011 and has not yet been returned. The DVD was bought from HiFi Suppliers of 14 Montague Street, Dublin for €1 on 1st Jan 2011. (10 marks)



Description: When a student gets an account (StudentAccount) he / she must provide a valid student number (StudentNumber). Staff members are recorded with their staffNo a name (SName) and may provide an e-mail address. All pieces of hardware owned are recorded with their type (HWType) (e.g. mouse) and may have a location (Locatn). When a student reports a problem, he / she must give a student number (StudentNumber), the serial no (HWSerialNo) of the piece of hardware that has the problem and the nature of the problem (NatureOfProblem). When the problem is recorded, the system date is recorded as PDate. A member of staff is allocated to each problem (i.e. a staffno is put into the problem row to show who is responsible for solving it). When the problem is solved, the staff member who solves it updates it, giving the date it was solved (SolvedDate). (ERD shown above)

Case Study Problem tracking system – ERD and Description

2. (a) Write a query to return the StudentAccount, Hardware Serial number (HWSerialNo), NatureofProblem, problem date (Pdate) and the solved date for all solved problems that took more than 2 days to fix. (10 marks) (5 marks)

(b) Write a query to return each member of staff’s StaffNo and the number of queries he / she solved. (10 marks)

(c) Write a query to return the staff number (Staffno) and number of problems solved for any member of staff who solved more than 5 problems. (10 marks)

3. (a) Write a query to return the StudentNumber, NatureOfProblem and Date the problem occurred (Pdate) of all problems reported since ’01-JAN-2011’ (10 marks)

(b) Write a query to return the studentnumber and studentaccount of all students who never reported a problem. (10 marks)

(c) Write a query to return the studentnumber and studentaccount of all students who reported a problem with a hardware item that has a HWtype ‘KEYBOARD’. (10 marks)

4. (a) Write an SQL statement to add a problem to the system, reported by the student with StudentAccount POBYRNE on an item of hardware with serial no DEL\_10111, reporting the problem as ‘VGA cable faulty’, using the current date as PDate. (10 marks)

(b) Update the problem previously added to give the responsibility of solving it to the staff member with staff number 12345.

(c) Combine the above into a single *transaction* (not a single statement), making it permanent before exiting. (10 marks)

**ALTER TABLE *tablename***

**{[ADD | MODIFY] *column-definition*}**

**CREATE TABLE *tablename***

**({*column-definition*}**

**[PRIMARY KEY ({*column-name},*)]**

**{[FOREIGN KEY ({*column-name}*) REFERENCES *tablename*]})**

**COMMIT**

**DELETE FROM *tablename* WHERE *condition***

**DROP [TABLE *tablename*|DROP VIEW *viewname*]**

**INSERT INTO *tablename* [*{column-name,}*] VALUES (*data-value-list*)**

**ROLLBACK WORK**

**SELECT *column-list* FROM *tablename***

**[WHERE *condition*]**

**[ORDER BY *column-list*]**

**[GROUP BY *column-name*]**

**[HAVING *condition*]**

***Conditions : =,>,<,>=,<=,<>,* BETWEEN .. AND.., IN *(list),* IS NULL, LIKE**

***Logical operators:* AND, OR, NOT**

***Set operations:*  UNION, MINUS, INTERSECT**

**SELECT SYSDATE FROM DUAL;**

**SELECT column-list FROM**

**<tablelist> WHERE <column-list> IN | ALL |ANY |EXISTS (Select statement)**

**UPDATE *tablename***

**[SET *column-name=* <*data-value*>] [WHERE *condition*]**

***Column-definition = column-name* [CHAR [(*n*)] | VARCHAR(*n*) | NUMBER [ *n*,*p*] | DATE | DATETIME] {[NOT NULL | UNIQUE | PRIMARY KEY]}**