DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science

DT282 BSc. (Honours) Degree in Computer Science (International)

Year 2

WINTER EXAMINATIONS 2016/2017

DATABASES I [CMPU2007]

MS. DEIRDRE. LAWLESS
DR. DEIRDRE. LILLIS
MR. KEVIN FOLEY

TUESDAY 10TH JANUARY 9.30 A.M – 11.30 A.M.

ANSWER ALL QUESTIONS.

THERE IS A SYNTAX TABLE ON THE LAST PAGE TO ASSIST YOU.

Case Study: Linda's Lighting (Use in Q1)

The following relational schema and interpretation will be used in subsequent questions:

client(cID, cName, cAddress, cEmail)
designer(designerID, dName, dEmail, dRateofPay)
product(prodID, prodDesc, prodPrice)
specification(specID, clientID, designerID, specDesc, specPrice)
specProduct(specID, prodID)
Note: underlined attributes indicate how each instance of a relation is identified.

A lighting shop Linda's Lighting Emporium offers a service whereby they will design a lighting system to the client's specifications and sells them the products needed for that system. The shop specialises in all types of lighting – indoor and outdoor, commercial and domestic.

The shop employs a number of designers. Details of designers are stored in the `designer` table and include the unique identifier (`designerID`), their name (`dName`), their email address (`dEmail`) and their hourly rate of pay (`dRateofPay`).

Details of clients are stored in the `Client` table. Stored for each client is a unique identifer (`cID`), their name (`cName`), their address their name (`cAddress`) and email address (`cEmail`).

Details of lighting products sold by the shop are stored in the `product` table. Stored for each team are its unique number (`prodID`), description (`prodDesc`), and price of the item (`prodPrice`).

Specifications are created for clients by designers. Details are stored in the `specification` table. Each specification has a unique identifier (`specID`). Also stored are the identifier of the client commissioning the system (`clientID`), the designer designing it (`designerID`), the date on which the specification was commissioned (`specDate`) plus a price (`specPrice`).

One specification can use many products, each product can be used many specifications.These details are stored in the `specProduct` table. Each instance is represents a single product used on a specification, identified by the combination of the identifier of the specification (`specificationID`) and the identifier of the product (`productId`).

1. (a) (i) Draw a *physical Entity Relationship diagram (ERD)* for the case study above showing the *attributes* for each entity, the *relationships* between them and identifying clearly *primary keys* and *foreign keys.*

Note: You DO NOT need to indicate attribute datatypes on this diagram.

(15 marks)

(ii) Explain *what type of entity* `specProduct` is. Explain clearly how the relationship between `specification` and `product` would be represented on a logical ERD and why `specProduct` is needed on the physical ERD.

(5 marks)

1. (b) When creating Linda's Lighting database the following requirements must be adhered to :

- all identifiers must be numeric capable of storing up to 6 digits;
- all names, addresses and descriptions must allow alphanumeric values up to 50 characters and cannot be null;
- products must have a price up to 99.99 but must be less than 72.50;
- specification prices can be up to 99,999.99;
- designers hourly rate of pay can store values up to 99.99;
- clients and designers need to provide an email address which must allow alphanumeric values up to 30 characters which must contain the @ symbol;
- designers email addresses must be unique;
- primary key constraints must be named with the name of the table and _pk and must be created at table level;
- foreign key constraints must be named with the names of the tables involved and _fk e.g. table1_table_2_fk and must be created at table level;
- all check and unique constraints must be created at table level.

(i) Write the SQL to *Create* the tables required by Linda's Lighting database *including all constraints required.* Explain the order in which the tables need to be created and why.

(15 marks)

(ii) Explain clearly what *type of data integrity* each of the constraints you implemented (including primary keys) helps you to achieve.

(10 marks)

Q1 Continues on Next Page

1. **(c)** Suppose you have created the tables and no data has yet been inserted into any of the tables. You attempt to execute the following insert statements in the order they are written below:

```
Insert into specProd values (101, 99);

Insert into specProd values (102, 99);

Insert into specProd values (101, 98);

Insert into specification values (101, 101, 101, 'A Simple
Lighting System', 900.00);

Insert into specification (specId, clientId, designerId,
specDesc, specPrice) values (102, 101, 101, 'A Garden
Lighting System', 1200.00);

Insert into specification values (specId, clientId,
designerId, specDesc, specPrice) (101, 'Mr. Grumpy, 'Small
Brown House, Dillydale', 'atop@gmail.com');

Insert into client (cId, cNAme, cAddress) values (102, 'Ms.
Brainy', 'The Zoo, Dillydale');

Insert into client (cId, cNAme, cAddress) values (101, 'Ms.
Helpful', 'The House on the Green, Dillydale',
'mhelp@gmail.com');

Insert into designer values (101, 'G. Versacci',
'gv@gmail.com', 70.00);

Insert into designer values (102, 'Donaltella Versacci',
'gv@gmail.com', 75.00);

Insert into product values (99, 'Chandelier', 80.00);

Insert into product values (98, ' Lava Lamp', 55.49);
```

Identify the errors that exist in the SQL given above and how you would correct these errors.

(12 marks)

(d) How would you *persist* the data inserted by the statements in part (c)? Why do you need to?

(3 marks)

2. Suppose we have cleared the database and inserted new data as follows:

Client

CID	CNAME	CADDRESS	CEMAIL
101	Mr. Bump	The hospital, Dillydale	mbump@gmail.com
102	Ms. Tidy	Neat Cottage, Dillydale	mtidy@gmail.com

Product

PRODID	PRODDESC	PRODPRICE
1	Chandelier	70
2	Solar Tree	30
3	Fairy Lights	20

Designer

DESIGNERID	DNAME	DEMAIL	DRATEOFPAY
111	C. Chanel	cchanel@email.com	87.5
112	G. Gucci	ggucci@email.com	67.5

Specification

SPECID	SPECDESC	SPECPRICE	CLIENTID	DESIGNERID
1	Fashion Lauch Lighting	7000	102	112
2	Halloween Lighting	900	101	111

SpecProd

SPECID	PRODID
1	1
1	2
1	3
2	2
2	3

(a) Write the SQL to retrieve the `designer name`, `client name`, `specification description` and `price`. Format the output so that the names of the columns appearing in the output are Client, Designer, Specification, Price and include the Euro symbol before the price.

(8 marks)

(b) Write the SQL to add a column to the `specification` table called `bonus` which can take numeric values up to 99999.99.

Populate this column using a *sub*-query which sets the value for to be equal to 10 times the hourly rate of pay of the designer who worked on the specification. Make sure that this column has a value > 0.

Include the SQL needed to retrieve all values from the `specification` table after the update is complete.

Hint: More than one SQL statement will be required.

(8 marks)

(c) Explan how the SQL you wrote for parts (a) and (b) works and present the output you would expect when the SQL is executed.

(2 x 2 marks)

3. Suppose the data in the database is as given in Q2 (a).

(a) Write SQL to to calculate the `total price` for services used in each booking by each guest. Restrict the output to only include specifications with a cost > 50.

Format your output to follow this template:

```
'The number of services used in booking <bookingID> by guest <guestId>
is <no. of services> and the total cost of these services is  <sum of
service price>'
```

Hint: You do not include the < > in your output. Retrieve bookingID, calculate the no. of services and the sum of the price.

(8 marks)

(b) Explain how the SQL you wrote for parts (a) works and present the output you would expect when the SQL is executed.

(3 marks)

(c) Suppose that we have successfully added a new designer C. Gucci into our designer table but that this designer has not yet worked on any specifications. A SELECT statement involving the `designer` and `specification` table is executed and produces the following output:

DNAME	Specification Description	Cost of Specification
G. Gucci	Fashion Lauch Lighting	7000
C. Chanel	Halloween Lighting	900
C. Gucci	No specifications completed	0

Write the SQL to achieve the output shown and explain how it works.

Hint: Think about what is in the designer table and what is in the specification and how to handle null values.

(9 marks)

SYNTAX TABLE

```
ALTER TABLE [schema.]table column_clauses;
 column_clauses:
   ADD (column datatype [DEFAULT expr] [column_constraint(s)] [,...] )
   DROP COLUMN column
      [CASCADE CONSTRAINTS] [INVALIDATE] CHECKPOINT int
   MODIFY column datatype [DEFAULT expr] [column_constraint(s)]
   RENAME COLUMN column TO new_name
ALTER TABLE [schema.]table constraint_clause [,...];
constraint_clause:
   DROP PRIMARY KEY [CASCADE] [{KEEP|DROP} INDEX]
   DROP UNIQUE (column [,...]) [{KEEP|DROP} INDEX]
   DROP CONSTRAINT constraint [CASCADE]
   MODIFY CONSTRAINT constraint constrnt_state
   MODIFY PRIMARY KEY constrnt_state
   MODIFY UNIQUE (column [,...]) constrnt_state
   RENAME CONSTRAINT constraint TO new_name
CREATE TABLE table (
      column datatype [DEFAULT expr] [column_constraint(s)[,...]] [,column
datatype [,...]]
         [table_constraint [,...]])
COMMIT
DELETE FROM tablename WHERE condition
DROP [TABLE tablename|DROP VIEW viewname]
INSERT INTO tablename (column-name-list) VALUES (data-value-list)
ROLLBACK
SELECT [DISTINCT] select_list
      FROM table_list
         [WHERE conditions]
               [GROUP BY group_by_list]
                  [HAVING search_conditions]
                     [ORDER BY order_list [ASC | DESC] ]
Conditions:=,>,<,>=,<=,<>, BETWEEN .. AND.., IN (list),IS NULL, IS NOT NULL,
LIKE
Logical operators:  AND, OR, NOT
Set operations:  UNION, MINUS, INTERSECT
CASE [ expression ]
   WHEN condition_1 THEN result_1
   WHEN condition_2 THEN result_2
   WHEN condition_n THEN result_n
   ELSE result
END
```

SYNTAX TABLE

SELECT

```
      ... FROM table1 LEFT JOIN table2
         ON table1.field1 compopr table2.field2 | USING clause
      ... FROM table1 RIGHT JOIN table2
         ON table1.field1 compopr table2.field2 | USING clause
      ... FROM table1 INNER JOIN table2
         ON table1.field1 compopr table2.field2| USING clause
```

Key

table1, table2	The tables from which records are combined.
field1, field2	The fields to be joined.
compopr	Any relational comparison operator: = < > <= >= or <>

UPDATE *tablename*
[SET *column-name= <data-value>*] [WHERE *condition*]
Column-definition = column-name **[CHAR [(*n*)] | VARCHAR2(*n*) | NUMBER [*n,p*] | DATE | DATETIME] {[NOT NULL | UNIQUE | PRIMARY KEY]}**

Oracle Functions

Null Handling Functions: NVL, NVL2, NULLIF, COALESCE, CASE, DECODE.
Case Conversion functions - Accepts character input and returns a character value: UPPER, LOWER and INITCAP.
Character functions - Accepts character input and returns number or character value: CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM and REPLACE.
Date functions - Date arithmetic operations return date or numeric values: MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND and TRUNC.
Group Functions: SUM([ALL | DISTINCT] expression); AVG([ALL | DISTINCT] expression); COUNT([ALL | DISTINCT] expression); COUNT(*); MAX(expression);MIN(expression)

down @ 10 am 1st

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF: W228/206C / W282/206C VENUE: B1

SUBJECT: Data Bases 1

DATE: 18/01/17

TIME: 9.30 - 11.30

SIGNED: [signature] for D. Lawless

INSTRUCTIONS:

Last line pg2

Should be

(specID) and re identifier of product (prodID).

2nd

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF: W228/206C
W282/206C

VENUE: B1, K106, K107, G33

SUBJECT: DATABASES I

DATE: 10/1/17

TIME: ~~10.00~~ 9.30 - 11.30

SIGNED: [signature]

INSTRUCTIONS:

6 lines from end pg 2
should read a description of the
system (spec Desc)

NOT the date.

3rd

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF: 228/206C, 282/206C VENUE: B1, K107, K106, G33

SUBJECT: Databases 1

DATE: 10/01/17

TIME: 9.30 – 11.30

SIGNED: Eric. Mahr Signing Per Deirdre Cawless.

INSTRUCTIONS:

(6)

point (3)

Product price can accommodate values up to 99.99 but the value should be < 72.50.

4.11

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF: W228/206C, W282/206C VENUE: B1, K106, K107, G33

SUBJECT: Databases 1

DATE: 10/01/17

TIME: 9.30 - 11.30

SIGNED: Rutherford

INSTRUCTIONS:

Q 3a

Write the SQL to calculate the total cost for products used in each specification ~~by each client~~.

Format your output to follow the template

The number of products used in specification <specID> ~~by client~~ is <# of products> and the total cost is <sum of productprice>.