

TECHNOLOGICAL UNIVERSITY DUBLIN
KEVIN STREET CAMPUS

BSc. (Honours) Degree in Computer Science (Infrastructure)

BSc. (Honours) Degree in Computer Science

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

Year 3

WINTER EXAMINATIONS 2020/21

Databases 2

Ms. Patricia O'Byrne

Dr. Deirdre Lillis

Dr. David Malone – DT211C

Dr. Martin Crane – DT228/DT282

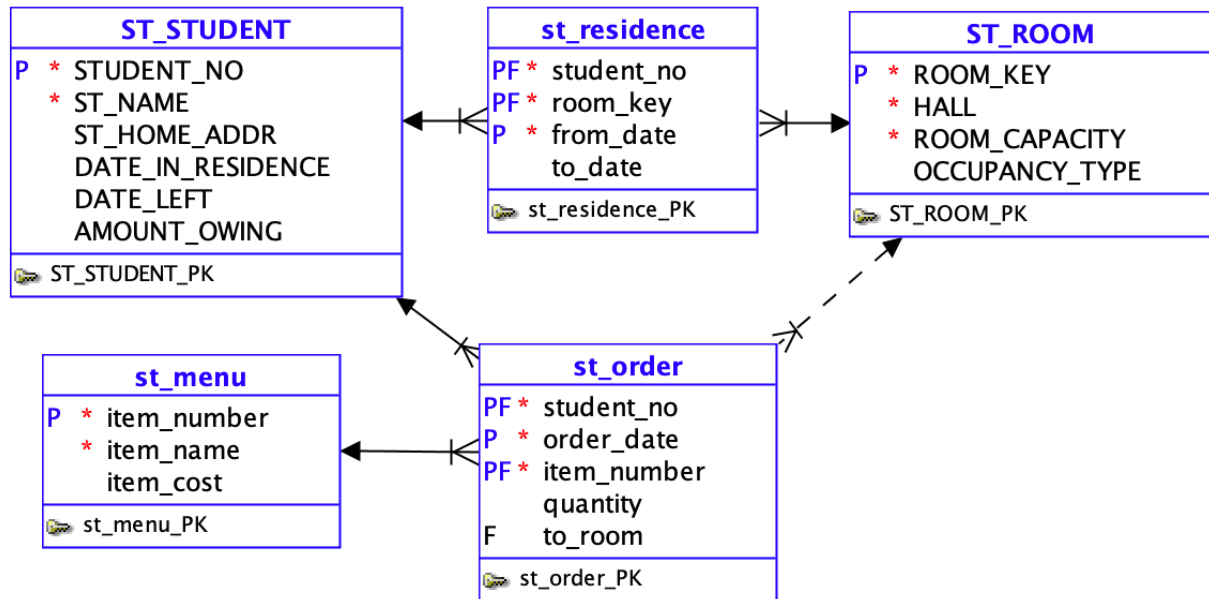
Two Hours

Instructions to candidates:

Please read Case Study 1 carefully before answering any questions.

Please answer question 1 and any two other questions.

There is a syntax sheet at the end of the paper to assist you.



A university runs a residence on the college campus, for its students. Students who are studying in the college can rent rooms in the residence. There are different types of apartments; some have individual rooms with en-suites, but share a kitchen and living area between four or six students. Some are for couples and these have their own apartment. Undergraduates are in a different wing to postgraduates. Every student has access to a kitchen, but there is also on-site catering, that can either be collected or delivered to a room. There is a standing charge of €200 at the start of the residency, giving the student an AMOUNT_OWING amount of -€200. Students can go into arrears of €50. When this is used up, students cannot order any more until they pay. The residence records each order and students in arrears are sent a bill at the end of each month. The record of orders is kept until the end of the academic year.

Case Study 1 Student residence

Question 1 (compulsory)

[40 marks]

1. (a) Prospective students can see the rooms that are available for rent. Student residents can see all their own details. They can see the full menu and can add orders online. Caterers can add and delete menu items. They can see and can update orders to indicate that they are delivered. Porters can add students and allocate students to rooms.

Copy the table below into your examination script and using the information described in Case Study 1, fill in the boxes with permissions (S for select, I for insert, U for update, D for delete) that each type of user has over the tables, using the principle of least privilege. **(6 marks)**

	Student	Caterer	Porter
ST_STUDENT			
ST_ROOM			
ST_MENU			
ST_RESIDENCY			
ST_ORDER			

- (b) Write queries to do the following: (3x6 marks)

(i) Write SQL to return a list of all item numbers (item_number), item_name and the total quantity of that item that has been ordered. **(6 marks)**

(ii) Write SQL to list the names of students who have ordered the menu_item with item_name 'Lasagne'. **(6 marks)**

(iii) Write SQL to list the room-key and number of occupants for any room that has fewer occupants than its capacity, ROOM_CAPACITY (i.e. there's a place left). **(6 marks)**

- (c) Write a PL/SQL function total_order_cost to take input parameters of date and student_no and return the total amount that student has spent on orders since the date. **(10 marks)**

- (d) Assuming the conceptual schema shown is in a schema named SRESIDENCE, show how you would make it possible for a user with the schema name JBLOGGS to run a query to return the student_name, amount_owing and total_order_cost for the student with student_no 'C11223344' in the last thirty days, using the function written in part (c). **(6 marks)**

Question 2

[30 marks]

2. Violations of residence rules are recorded. Undergraduate resident students are subject to curfews and must be back in their room by midnight during the working week. Undergraduates who are late must be admitted by one of the residence porters, who records a violation of the curfew. Each apartment is inspected every month. Following the inspection, if the porter finds damage, he / she photographs the damage and records a rules violation against the student. Photographs are named with the student number, date and time of photo. A third type of violation results if a student's behaviour is not appropriate, when an 'unruly' violation is recorded against him or her. *Note: Given that several staff are working simultaneously, two violations could be recorded at the same time, by different staff members.*

Front desk log				Date:		31/10/2019			
General comments:			Due to the Halloween celebrations, there complaints of noise from the post-graduate wing. Fireworks were confiscated from a few undergraduates.						
Time	Porter	Name	Student No.	Name	Room key	Violation Type	Description	Evidence	Action
18:00:00	FD1003	Jacob Greene	C11223344	Gina Nolan	N401	Damage	Detonated fireworks in the room, causing scorch marks	Photograph 'C11223344201910311800.png'	Violation recorded, Referred to Maintenance
18:10:00	FD1003	Jacob Greene	C11223345	Tara Fox	N401	Unruly	Abusive to staff investigating explosive noise.	Staff report	Violation recorded
00:40:00	FD1003	Jacob Greene	C11223348	Jacques Verne	E408	Late	Arrived back at 12:40am	Staff report	Violation recorded
14:00:00	FD1007	Marion Gleeson	C14311122	Peter Jones	E331	Damage	Left room in very bad state. Build up of food waste.	Photograph 'C14311122201910311400.png'	Violation recorded, Referred to Maintenance
18:00:00	FD1007	Marion Gleeson	C11223348	Tara Fox	N401	Damage	Left room in very bad state. Build up of food waste.	Photograph 'C11223348201910311800.png'	Violation recorded, Referred to Maintenance
00:20:00	FD1007	Marion Gleeson	C18122211	Pamela Dunne	W220	Late	Arrived back at 12:20am	Staff report	Violation recorded
00:22:00	FD1007	Marion Gleeson	C18122218	James Dunne	W220	Late	Arrived back at 12:20am	Staff report	Violation recorded

Figure 1 Front desk log from Student Residence, for 30th October 2019

- Represent the information supplied in Figure 1 in unnormalized form (5 marks)
- Represent the information in first normal form (5 marks)
- Represent the information in second normal form (5 marks)
- Represent the information in third normal form (5 marks)
- Draw a fully normalized ERD to represent the entities derived by the normalization process, showing primary and foreign keys, attributes and relationships. (10 marks)

Question 3

[30 marks]

3. (a) Using the ERD shown in Case Study 1, write SQL to create a view `FILL_ORDERS` of the data that the catering staff need when making up orders. The caterer needs to see `student_no`, `st_name`, `to_room`, `hall`, `order_date`, `item_number`, `item_name`, `quantity` and `item_cost` for all student orders where the student `amount_owing` is less than €50 and the item is available (`item_availableYN='Y'`). **(10 marks)**
- (b) Design a MongoDB collection or set of collections that can hold information from the following orders from two separate students, observing the rule that every order must at least have a `student_number`, `item_number`, and an `order_date`. Write code to insert one of these orders:
- 1) Pamela Dunne (C18122211) , who owes €20 ordered 2 `item_number` 30, Green Tea at a cost of €1 each, and 2 `item_number` 14, Veggie Burger, at €5 each, to be delivered to room W220 in hall “North Wing 2nd Floor” on 10th Oct 2019 at 18:00.
- 2) Tara Fox (C11223345) who owes €50 ordered 1 `item_number` 1, Coffee (Black, strong, no sugar please) and 1 `item_number` 2, Porridge (two teaspoons of sugar please) to be delivered to room N401 in hall “North Wing 4th Floor” on 10th Oct 2019 at 11:00. **(10 marks)**
- (c) Discuss your preference for how this data should be stored, giving reasons. **(10 marks)**

Question 4

[30 marks]

4. Assuming the data is stored in a relational database with the conceptual schema as shown in the ERD in Case Study 1, write a PL/SQL program, complete with error checking, to add an extra order in the following steps:
- (a) Using substitution variables, accept a `student_no`, `order_date`, `item_number`, `quantity` and `room_key`. **(4 marks)**
- (b) Verify that the student exists and owes less than €50. If this is not the case, report that the student is not eligible to order and exit without amending the data. **(7 marks)**
- (c) Verify that the item is on the menu and is available (`item_availableyn = 'Y'`). If this is not the case, report that the item is not available and exit without amending the data. **(7 marks)**
- (d) Add the new order, with the value 'N' for `DeliveredYN`, reporting that the order has been placed, making all changes persistent, handling errors that may arise. **(12 marks)**

NOTE: Reporting implies displaying the message on the screen.

SQL

SELECT column-list **FROM** tablename

[**WHERE** condition]

[**ORDER BY** column-list]

[**GROUP BY** column-name]

[**HAVING** condition];

SELECT column-list **FROM** join-expression

Join-expression =

table1 **JOIN** table2 **ON** condition /
USING (column-list)

table1 **LEFT JOIN** table2 **ON**
condition / **USING** (column-list)

Conditions : =,>,<,>=,<=,<>,

BETWEEN .. **AND**.., **IN** (list),
IS NULL, **LIKE**

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

Set operations: **UNION**, **INTERSECT**,
EXCEPT

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

UPDATE tablename

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

PL/SQL_BLOCK

DECLARE

[constant/variable declarations]

BEGIN

Executable statements

RETURN Return value

[**EXCEPTION**

exception handlers

END;

Exceptions:

DUP_VAL_ON_INDEX

NO_DATA_FOUND

TOO_MANY_ROWS

Error variables:

SQLCODE – gives error code

SQLERRM – gives error message.

Syntax provided depends on the platform used during delivery of the module.

MONGODB EXAMPLES

Create a products collection:

```
db.createCollection("contacts",
{ validator:{ $or:[
  {phone:{$type:"string"}},
  {email: {$regex: /@mytudublin\.ie$/}},
  {status:{$in:["Unknown","Incomplete"]}}
] ]}}
```

Insert a document with _id value 11 into products collection:

```
db.products.insert(
[
  { _id: 11, item: "pencil", qty: 50, type:
"no.2" },
  { item: "pen", qty: 20 },
  { item: "eraser", qty: 25 }
])
```

Attributes may contain embedded documents or arrays.

<Embed_attrrib_name>:[{set of embedded attributes