

**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**

---

SEMESTER 1 EXAMINATIONS 2019/20

---

**Databases 2**

Ms. Patricia O'Byrne  
Dr. Deirdre Lillis  
Dr. David Malone – DT211C  
Dr. Martin Crane – DT228/DT282

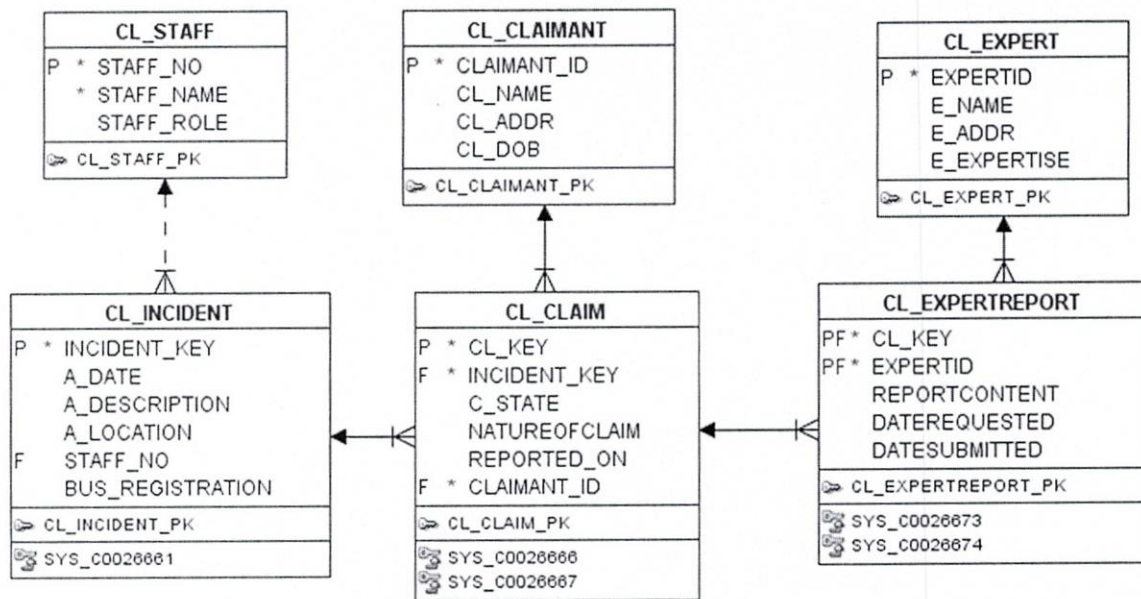
Two Hours

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 company owns a fleet of buses on which it carries fare-paying passengers. On occasion, incidents occur that result in a claim. The company's claims section handles most of the claims. When an incident occurs, staff members are expected to report it, giving his or her staff number (staff\_no), name (staff\_name) and role (staff\_role) along with details of the incident (a\_date, a\_description, a\_location). If a bus is involved, its registration is also recorded (bus\_registration). An incident key (incident\_key) is generated at the time of recording.

Claimants can submit claims, giving the incident details (a\_date, a\_description, a\_location), their details (cl\_name, cl\_addr), date of birth (cl\_dob) and the nature of the claim (natureofclaim). A claim key (cl\_key) is generated at the time of recording. If the incident has not yet been reported by a member of staff, it is recorded with the claim and the claim state (c\_state) is 'U' for unconfirmed. If a staff member confirms it, the state is updated to 'C' for confirmed and the staff\_no is added to the incident. While assessing the claim, the assessor may look for an expert report, creating the request (cl\_expertreport) giving the claim key (cl\_key) and picking an appropriate expert (expertid) from the experts table (cl\_expert). This action causes the claim state to change to 'E' for awaiting expert. When the expert has finished reviewing the claim, he / she updates the report, adding content (reportcontent) and the date he / she submitted the report. If all expert reports on the claim have been submitted, the claim state (c\_state) reverts to 'C' for confirmed. When the claim is assessed its state can change to 'S' for settled or 'R' for rejected.

### Case Study 1 Handling Insurance Claims

**Question 1 (compulsory)**

**[40 marks]**

1. (a) Claimants can make claims by electronic means, but can only ever see data specific to them. They cannot see incident details or details of other claims related to that incident. The assessor can see everything, and can change the claim state and request expert reports. The expert can see details of the claimant, the claim and the incident and can submit report content and dates submitted for expert reports requested of them and change their own address. Other staff members can see their own details and incidents and can verify them, by entering their staff no.

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) that each type of user has over the tables, using the principle of least privilege. **(6 marks)**

	Claimant	Assessor	Expert	Other staff member
CL_CLAIM				
CL_CLAIMANT				
CL_INCIDENT				
CL_STAFF				
CL_EXPERT				
CL_EXPERTREPORT				

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

(i) Write SQL to return a list of staff names (STAFF\_NAME) who are working as drivers (STAFF\_ROLE='Driver') who have never reported an incident. **(6 marks)**

(ii) Write SQL to list the names of claimants (CL\_NAME) whose claims have been investigated by a doctor (E\_Expertise is 'Medical'). **(6 marks)**

(iii) Write SQL to list the expert name, expertid and e\_expertise of any expert that has submitted expert reports on more than ten claims. **(6 marks)**

- (c) Write a function count\_claims(DOB) to take an input parameter of a date and return number of claims made by claimants who were born on that date, including error and exception checking. **(10 marks)**

- (d) Explain how this function could be called from a 3GL (like Python) in a safe and secure way, noting security you would apply and the steps required in a 3GL to retrieve data from a relational database. **(6 marks)**



**Question 2**

**[30 marks]**

2. The company has extended the amount of information they store. Evidence is stored in different electronic stores depending on what type of evidence it is – e.g. video evidence is always kept in the Video Store.

<b>Incident Key: In123</b> <b>Incident Description:</b> Low-speed collision between bus and car. Car passengers claim to suffer ongoing pain as a result. Car has some damage. Neither driver admits responsibility.								
Claim Key	Witness No	Witness Name	Witness Phone	Evidence Key	Evidence Type	Evidence Store	Evidence File	Evidence Description
CL123	423	Mary O'Connell	086 332211	EV123	Video	Video Store	V123.MP4	Video footage of scene from GoPro.
CL123	583	Halligan's Deli	(01) 44334	EV124	Video	Video Store	V134.MP4	Video footage of scene, CCTV.
CL123	43	Dr. H. Macken	(01) 4455445	EV125	Medical	Medical Reports	V125.pdf	Med Report on J.K.'s injuries
CL124	43	Dr. H. Macken	(01) 4455445	EV126	Medical	Medical Reports	V126.pdf	Med Report on M.K.'s injuries
CL124	86	Mike Jones	(01) 3322332	EV127	Mechanical	Mechanic Store	V127.pdf	Report on damage to vehicle.

**Table 1 Progress report on claims arising from incident with key In123**

- Represent the information supplied in Table 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 of the data that the assessor needs, when assessing a claim. The assessor should see claims where the incident has been confirmed, but not those that have been settled. He / She should see all claimant, claim and incident information, all details on staff members who have reported or confirmed the incident and expert reports that have been requested on the claim. Remember that not all claims have an expert report. **(10 marks)**

- (b) Design a MongoDB collection or set of collections that can hold information from the following two incidents, observing the rule that every claim must have a claimant with contact details and an incident report, with a location. Write code to insert one of these incidents:

Incident A: happened on 20th Nov 2019, at the corner of North and Barrow Street, Latitude: 53° 21' N, Longitude: -6° 19' W. Involved Bus (reg 151 D 12345). Incident Report: "Bus pulled out and clipped the front wing of my car (192 D 1001)." Claim: Kenneth Lynch, DOB 12<sup>th</sup> Jan 1993, 23 Olive Grove, D55 is claiming for compensation for vehicle damage and replacement vehicle hire.

Incident B: Happened on 21<sup>st</sup> Nov 2019, on the 43 bus, 152 D 44332. "I was wearing a new pair of trousers and sat in the third row of seats in the bus. When I stood up there was chewing gum on my trousers. I told the driver and he said he'd noted it." Claim: "I, Simon McGlynn, 87B Johnsway House, Johnsway Lane, D99, born on 12<sup>th</sup> Apr 1974, am claiming for replacement value of the trousers. I enclose the receipt for the trousers (receipt.png)." Confirmation: Ted McBride (staff no 77321, driver) confirmed this incident on 22<sup>nd</sup> November. **(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 claim against an existing incident, by an existing claimant, in the following steps:
- (a) Using substitution variables, accept a claimant\_id, an incident\_key and the nature of claim (natureofclaim, a string). **(5 marks)**
  - (b) Verify that the claimant has an existing claim for this incident, noting the state of the existing claim. If there is no such claim, report this and exit without amending the data. **(15 marks)**
  - (c) If the claimant has an existing claim for this incident, add a new claim, incrementing an existing sequence called cl\_sequence and using the current date to populate the 'reported\_on' attribute. If the existing claim has a state of 'U', the new claim should also have a state of 'U'. Otherwise, the new claim has a state of 'C'. **(10 marks)**



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,  
MINUS*

*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;*

*PL/SQL FUNCTION  
CREATE [OR REPLACE] FUNCTION  
function\_name  
[(parameter [,parameter])] ]  
RETURN return\_datatype  
AS  
[declaration\_section]  
BEGIN  
executable\_section  
[EXCEPTION  
exception\_section]  
END [function\_name];*

*ERROR VARIABLES  
SQLCODE – gives error code  
SQLERRM – gives error message.*

MONGODB EXAMPLES

**Create a products collection:**

```
db.createCollection("contacts",  
{ validator: { $or: [  
  { phone: { $type: "string" } },  
  { email: { $regex: /@mydit\.ie$/ } },  
  { 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_attr_name>: [{set of embedded  
  attributes}]
```