Q1

(a) Describe what a junction or bridge table is in the context of database design.

A junction table is a table used to associate the row of one table with the row of another table. This table is usually used to solve a many to many relationship problem

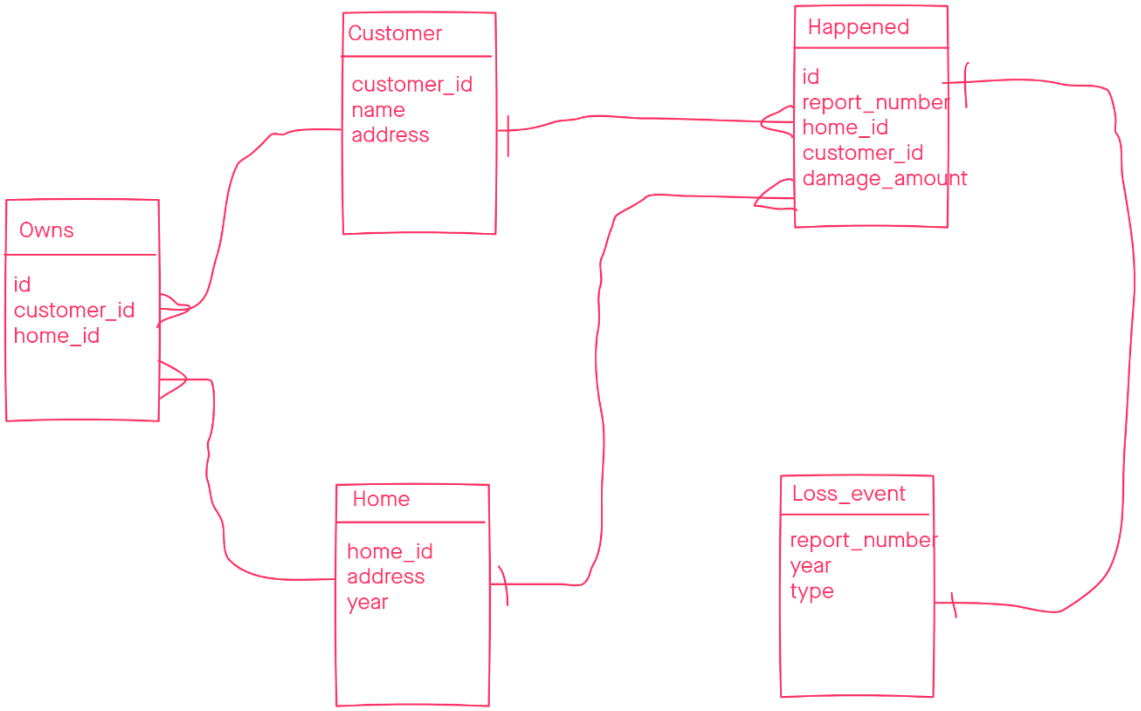
(b) When designing a database, which issues can arise when using many-to-many associations? Use an example and illustrations to clarify your answer. How can a junction or bridge table resolve the issues?

With a many to many association, the problem is that the row in the table could could contain many ids because it has many items related to it and so the table could become too large, thus a bridge table is used to map the row id of one table with the row id of another table

(c) Which table in the home insurance database fulfils the junction table role? Justify your answer.

Owns. It has two primary keys of two other tables

(d) Draw an Entity-Relationship diagram for the insurance database. Your diagram should include entities and relationships between entities and suitable association types.



(e) Write the SQL code to create the Owns table. Make sure it includes suitable
properties to represent the concept that a Customer owns a Home.

```
CREATE TABLE Owns (id INT AUTO_INCREMENT,
                   customer_id INT,
                   home_id INT,
                   PRIMARY KEY (id),
                   FOREIGN KEY (customer_id) REFERENCES Customer (customer_id),
                   FOREIGN KEY (home_id) REFERENCES Home (home_id));
```

Q2

(a) Write a SQL statement that finds the total number of loss events in and after 2020.

SELECT COUNT(report_number) FROM Loss_event WHERE year >= 2020;

(b) State the purpose of a SQL JOIN statement and explain how it works.

A join is a clause that is used to combine rows from two or more tables based on related column between them

(c) Explain the difference between SQL JOIN, SQL RIGHT JOIN, and SQL LEFT JOIN.

Sql join: This join combines all the columns from the rows of two or more tables that are joined together

Sql right join: this join combines all the rows from table to the right of the join and only matching rows from the left

Sql left join: this returns all the rows from the left table and only matching records from the right table

(d) Considering the database specification above, write a SQL JOIN statement that finds the total amount of damage associated with events involving a customer named John Smith.

```
SELECT SUM(Happened.damage_amount)
FROM Happened
JOIN Customer
ON Happened.customer_id=customer.customer_id
```

Alternative option (Darren):

```
SELECT SUM(Happened.damage_amount)
FROM Happened
JOIN Customer
ON Happened.customer_id=Customer.customer_id
WHERE Customer.name = “John Smith”;
```

(e) Write a SQL statement to find all types of loss events where a customer named ‘John Smith’ was involved.

```
SELECT type FROM loss_event WHERE report_number IN ( SELECT report_number FROM Happened WHERE customer_id IN (SELECT customer_id FROM Customer WHERE name=’John Smith’))
```

Alternative option (Darren):

```
SELECT Loss_event.type
FROM Loss_event
JOIN Happened
ON Loss_event.report_number = Happened.report_number
WHERE Happened.customer_id IN (
        SELECT Customer.customer_id
        FROM Customer
        WHERE Customer.name = ’John Smith’
);
```

Question 3:

a) Which line of code would you change to change the console output of the web server (web server prompt)?

Line 25 - start the web server using the built in function listen

b) Which line of code starts to add a 'route' to your web application? How do you know it is this line?

Line 20

c) Why do you need to import body-parser (line 14) in your Node.js web server code?

Is this a typo? Importing is on Line 2 of index.js?

body-parser needed to access the data in the body of the POST request method.

d) Which line of code accesses data collected from a POST request? Which form-fields does it access?

Line 4 in middle ware script

Accesses: “category” and “name” form fields

Alternative option (Diego):

Accesses: “category”, “name” and “price” form fields

e) Describe what templating is in the context of web application development.

Templating separates the logic from presentation markup. It supports the progression from static webpages to dynamic webpages:

- insert variables in into the final output template at run time
- run programming logic at runtime and send final HTML to display

f) Write a template file using the EJS embedded javascript templating library. It should display all jewellery items passed as a parameter from the middleware to the template file. The template file should generate a complete HTML page, it should have a title using a suitable HTML tag, the incoming variable is called availableItems. Each jewellery item stored in the database has three properties of category, name, and price.

```
<!doctype html>
<html>
	<head>
		<title>Jewellery Items</title>
		metatags ??
	</head>
	<body>
		<h1>All Jewellery Items</h1>
		<ul>
			<% availableItems.forEach(function(item) { %>
			<li><%= item.category %>, <%= item.name %>, £ <%item.price%></li>
			<%})%>
		</ul>
	</body>
</html>
```

g) Write a piece of middleware code, with a route named ‘display_all_rings’ to access the template file written in part (f) and retrieve and display all jewellery items in the ‘ring’ category stored in the database. Your code should query the database to get all rings. Your code should handle the error condition where the database query fails. Your code should render a template file to display all retrieved items from the database.

I’m VERY unsure about this answer ! Does not access answer in part (f) ?

```
app.get(“/display_all_rings”, function(res, req){
        // Query database for all rings
        let sqlquery = ‘SELECT * FROM jewellery WHERE category = “ring“‘;
        // Execute query
        db.query(sqlquery, (err, result) => {
                if(err){
                        //re-direct to homepage on error
                        res.redirect(“/”)
                }
                        //render template file (called itemQuery.ejs) to display retrieved items
                        res.render(“itemQuery.ejs”, {availableItems: result})
        });
});
```

h) You have been assigned a new project to develop a dynamic web application for a hospital records system using Node.js and MySQL. What are the steps you would follow as a full-stack developer for the development of this web application? Name at least two main steps for the design and two main steps for the implementation of the web application considering what you have learned in this module.

DESIGN

- Agree objectives and functionality with stakeholders ?

- Relational Database design (define schema / datatypes / tables / relations)
-

IMPLEMENTATION

- Selection of web hosting platform (cloud vs stand alone; security; cost, etc)
- REconfirm with users (optimise)
- Testing before deployment (especially midterms….)
- Maintenance and updates ?
- Form validation
- Security protocols ? HTTPS vs HTTP (application layer)