

Question	Answer	Marks								
7(a)	<p><b>1 mark</b> per bullet point to <b>max 3</b></p> <ul style="list-style-type: none"><li>• Flat-file has more data redundancy</li><li>• ... because the same data is stored many times // data is stored in different tables which are linked</li><li>• There is program-data dependence with flat-files</li><li>• ... because any changes to the structure of the data means the programs that access that data have to be re-written</li><li>• Flat-file has more data inconsistency // worse data integrity</li><li>• ... because duplicated data might be stored differently //...because when data is updated in one place, it is not updated everywhere</li><li>• It is not easy to perform <b>complex</b> searches /queries</li><li>• ... because a new program has to be written each time</li><li>• Flat files could have a lack of privacy</li><li>• ... as user views cannot easily be implemented</li></ul>	<b>3</b>								
7(b)(i)	<p><b>1 mark</b> for each correct example</p> <p>one-to-one</p> <ul style="list-style-type: none"><li>• e.g. customer to payment details // customer to login details</li></ul> <p>one-to-many</p> <ul style="list-style-type: none"><li>• e.g. customer to order</li></ul> <p>many-to-many</p> <ul style="list-style-type: none"><li>• e.g. order to product // customer to product</li></ul>	<b>3</b>								
7(b)(ii)	<p><b>1 mark</b></p> <table border="1"><thead><tr><th>Relationship</th><th>Tick (✓)</th></tr></thead><tbody><tr><td>one-to-one</td><td></td></tr><tr><td>one-to-many</td><td></td></tr><tr><td>many-to-many</td><td>✓</td></tr></tbody></table>	Relationship	Tick (✓)	one-to-one		one-to-many		many-to-many	✓	<b>1</b>
Relationship	Tick (✓)									
one-to-one										
one-to-many										
many-to-many	✓									
7(b)(iii)	<p><b>1 mark</b></p> <p>CREATE DATABASE SHOPORDERS ;</p>	<b>1</b>								
7(c)	<p><b>1 mark</b> per item to <b>max 3</b></p> <ul style="list-style-type: none"><li>• table name</li><li>• field name // attribute</li><li>• data type</li><li>• type of validation</li><li>• Primary Key</li><li>• Foreign Key</li><li>• relationships</li></ul>	<b>3</b>								

Question	Answer	Marks								
1(a)	<p><b>1 mark</b> for definition, <b>1 mark</b> for appropriate example in each</p> <table><tr><th>Term</th><th>Definition and example</th></tr><tr><td>Field</td><td>A column/attribute in a table e.g. CustomerID in the table CUSTOMER</td></tr><tr><td>Entity</td><td>Anything that data can be stored about e.g. A customer or a house</td></tr><tr><td>Foreign Key</td><td>A field in one table that is <b>linked</b> to a <b>Primary Key</b> in another table e.g. CustomerID / HouseID <u>in table RENTAL</u></td></tr></table>	Term	Definition and example	Field	A column/attribute in a table e.g. CustomerID in the table CUSTOMER	Entity	Anything that data can be stored about e.g. A customer or a house	Foreign Key	A field in one table that is <b>linked</b> to a <b>Primary Key</b> in another table e.g. CustomerID / HouseID <u>in table RENTAL</u>	6
Term	Definition and example									
Field	A column/attribute in a table e.g. CustomerID in the table CUSTOMER									
Entity	Anything that data can be stored about e.g. A customer or a house									
Foreign Key	A field in one table that is <b>linked</b> to a <b>Primary Key</b> in another table e.g. CustomerID / HouseID <u>in table RENTAL</u>									
1(b)	<p><b>1 mark</b> per bullet point to <b>max 2</b></p> <ul style="list-style-type: none"><li>• All fields in all tables are dependant fully on the PK and on no other fields</li><li>• for example all fields in Customer table are fully dependent on CustomerID</li></ul>	2								
1(c)(i)	<p><b>1 mark</b> for each correctly completed line</p> <pre>CREATE TABLE RENTAL (     RentalID INTEGER NOT NULL,     CustomerID INTEGER NOT NULL,     HouseID <b>VARCHAR</b> (5) NOT NULL,     MonthlyCost <b>REAL/CURRENCY</b> NOT NULL,     DepositPaid BOOLEAN NOT NULL,     <b>PRIMARY KEY</b> (RentalID) );</pre>	4								
1(c)(ii)	<p><b>1 mark</b> per bullet point</p> <ul style="list-style-type: none"><li>• Select FirstName and LastName</li><li>• From both tables</li><li>• Where DepositPaid = No</li><li>• Joining tables (either AND, or INNER JOIN)</li></ul> <p><b>Example script:</b> SELECT FirstName, LastName FROM CUSTOMER, RENTAL WHERE DepositPaid = No AND RENTAL.CustomerID = CUSTOMER.CustomerID;</p>	4								

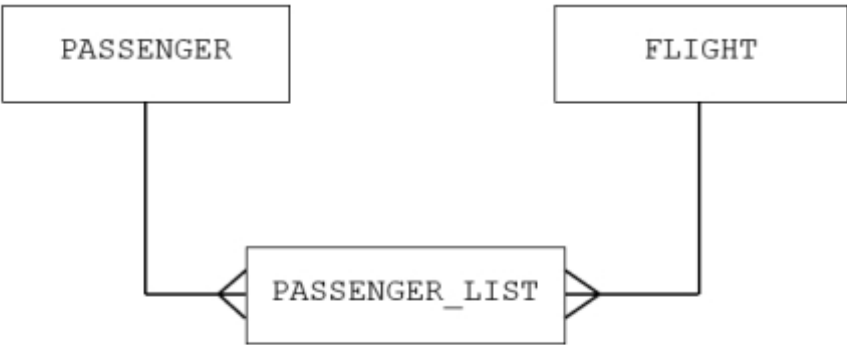
Question	Answer	Marks																				
5(a)	<p><b>1 mark</b> for 2 or 3 correct ticks, <b>2 marks</b> for 4 correct ticks</p> <table><tr><th>Table</th><th>Field name</th><th>Primary Key (PK)</th><th>Foreign Key (FK)</th></tr><tr><td>MANAGER</td><td>ManagerID</td><td>✓</td><td></td></tr><tr><td>SHOP</td><td>ManagerID</td><td></td><td>✓</td></tr><tr><td>CAR</td><td>RegistrationNumber</td><td>✓</td><td></td></tr><tr><td>CAR</td><td>ShopID</td><td></td><td>✓</td></tr></table>	Table	Field name	Primary Key (PK)	Foreign Key (FK)	MANAGER	ManagerID	✓		SHOP	ManagerID		✓	CAR	RegistrationNumber	✓		CAR	ShopID		✓	2
Table	Field name	Primary Key (PK)	Foreign Key (FK)																			
MANAGER	ManagerID	✓																				
SHOP	ManagerID		✓																			
CAR	RegistrationNumber	✓																				
CAR	ShopID		✓																			
5(b)	<p><b>1 mark</b> per bullet point</p> <ul style="list-style-type: none"><li>• Access rights give managers / himself access to different elements</li><li>• ... by having different accounts / logins</li><li>• ... which have different access rights e.g. read only // no access / read / write</li><li>• Specific <u>views</u> can be assigned to himself and to the managers</li><li>• ... e.g. managers can only see the data for their own shop(s)</li></ul>	3																				
5(c)(i)	<p><b>1 mark</b> per correctly completed statement</p> <pre>SELECT COUNT(RegistrationNumber) FROM CAR GROUP BY ShopID</pre>	3																				
5(c)(ii)	<p><b>1 mark</b> for each correct statement</p> <pre>INSERT INTO CAR VALUES ("123AA", "Tiger", "Lioness", 10500, "12BSTREET")</pre>	2																				

Question	Answer	Marks
4(a)	<p><b>1 mark</b> per bullet point to <b>max 4</b></p> <p>e.g.</p> <ul style="list-style-type: none"> <li>• Reduced data redundancy // less repeated data</li> <li>• ...because each item of data is only stored once</li> <li>• Maintains data consistency // improves data integrity</li> <li>• ...changes in one table will automatically update in another</li> <li>• ... linked data cannot be entered differently in two tables</li> <li>• Program-data independence</li> <li>• ...changes to the data do not require programs to be re-written</li> <li>• Complex queries are easier to run</li> <li>• Can provide different views</li> <li>• ....so users can only see specific aspects of the database</li> </ul>	<b>4</b>
4(b)	<p><b>1 mark</b> for 3NF</p> <p><b>1 mark</b> per bullet for justification to <b>max 2</b></p> <ul style="list-style-type: none"> <li>• There are no repeated attributes // it is already in 2NF</li> <li>• Each field is fully dependent on the corresponding primary key // no partial dependencies</li> <li>• No transitive dependencies</li> </ul>	<b>3</b>

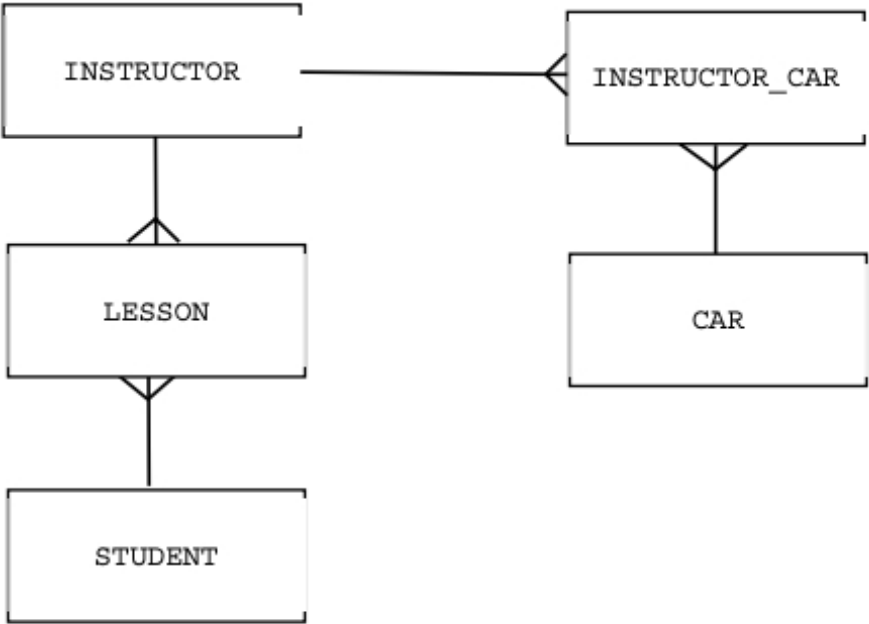
Question	Answer	Marks
4(c)(i)	<p><b>1 mark per bullet point</b></p> <ul style="list-style-type: none"> <li>• Create table, table name, opening and closing brackets</li> <li>• StudentID and Mark as integer</li> <li>• TestID as Varchar</li> <li>• Primary key correctly set up</li> <li>• Foreign keys correctly set up</li> </ul> <p>e.g.</p> <pre>CREATE TABLE STUDENT_TEST (     StudentID INTEGER,     TestID VARCHAR,     Mark INTEGER,     PRIMARY KEY(StudentID, TestID),     FOREIGN KEY(TestID) REFERENCES TEST(TestID),     FOREIGN KEY(StudentID) REFERENCES STUDENT(StudentID) );</pre>	<b>5</b>
4(c)(ii)	<p><b>1 mark for each point</b></p> <ul style="list-style-type: none"> <li>• AVG(Mark)</li> <li>• SELECT and FROM STUDENT_TEST</li> <li>• WHERE clause</li> </ul> <p>e.g.</p> <pre>SELECT AVG(Mark) FROM STUDENT_TEST WHERE TestID = "A7";</pre>	<b>3</b>

Question	Answer	Marks
5(a)	<p>1 mark for each link:</p> <ul style="list-style-type: none"> <li>• 1 CUSTOMER to many BOOKING</li> <li>• 1 COURSE to many BOOKING</li> <li>• 1 COURSE to many COURSE_EMPLOYEE</li> <li>• 1 EMPLOYEE to many COURSE_EMPLOYEE</li> </ul> <pre> graph LR     COURSE --&gt; BOOKING     COURSE --&gt; COURSE_EMPLOYEE     CUSTOMER --&gt; BOOKING     EMPLOYEE --&gt; COURSE_EMPLOYEE </pre>	4
5(b)	<p><b>1 mark</b> for description and <b>1 mark</b> for application to the given tables</p> <p>e.g. (2 marks)</p> <ul style="list-style-type: none"> <li>• each value stored in the CustomerID (FK) field in the Booking table must have a corresponding value (1) in the CustomerID (PK) field in the Customer table (1)</li> </ul> <p>e.g. (1 mark)</p> <ul style="list-style-type: none"> <li>• Each foreign key value must have a matching value in the primary key of the linked table (1)</li> </ul>	2
5(c)	<p><b>1 mark</b> each (<b>max 3</b>)</p> <p>e.g.</p> <ul style="list-style-type: none"> <li>• Tables</li> <li>• Fields/attributes</li> <li>• Indexes</li> <li>• Users</li> <li>• Primary Key</li> <li>• Foreign Key</li> <li>• Relationships</li> <li>• Views</li> </ul>	3

Question	Answer	Marks
5(d)	<p><b>1 mark</b> per syntactically correct bullet point:</p> <ul style="list-style-type: none"> <li>• Select correct fields</li> <li>• From correct table</li> <li>• Correct criteria for Role</li> <li>• Correct criteria for Language</li> </ul> <p><b>Example:</b>  SELECT FirstName, LastName  FROM EMPLOYEE  WHERE Role = "Leader"  AND (Language = "French" OR Language = "English");</p>	<b>4</b>

Question	Answer	Marks
9(a)	<p><b>One mark per correct relation</b></p>  <pre> graph TD     PASSENGER[PASSENGER]     FLIGHT[FLIGHT]     PASSENGER_LIST[PASSENGER_LIST]     PASSENGER === PASSENGER_LIST     FLIGHT === PASSENGER_LIST </pre>	<b>2</b>
9(b)	<p><b>One mark per bullet point</b></p> <ul style="list-style-type: none"> <li>• Create table <code>FLIGHT</code> with open and close brackets (and ;)</li> <li>• <code>FlightID</code> as <code>VARCHAR</code> restricted to max 6</li> <li>• <code>FlightDate</code> as <code>DATE</code></li> <li>• <code>FlightTime</code> as <code>TIME</code></li> <li>• Declaring <code>FlightID</code> as Primary Key</li> </ul> <p><b>Example</b></p> <pre> CREATE TABLE FLIGHT ( FlightID VARCHAR(6), FlightDate DATE, FlightTime TIME, PRIMARY KEY (FlightID)); </pre>	<b>5</b>



Question	Answer	Marks
7(a)	<p><b>1 mark per bullet point to max 2</b></p> <ul style="list-style-type: none"> <li>• Reduced data redundancy</li> <li>• Reduced data dependency</li> <li>• Improved data integrity</li> <li>• Improved data privacy</li> <li>• Program-data independence</li> <li>• Ability to create ad hoc queries</li> </ul>	<b>2</b>
7(b)	<p><b>1 mark for each correct link</b></p>  <pre> classDiagram     INSTRUCTOR --&gt; INSTRUCTOR_CAR     LESSON --&gt; STUDENT     INSTRUCTOR_CAR --&gt; CAR </pre>	<b>4</b>

Question	Answer	Marks
7(c)	<p><b>1 mark</b> for each correctly completed statement</p> <ul style="list-style-type: none"> <li>• CREATE (line 1)</li> <li>• INTEGER (line 6)</li> <li>• PRIMARY KEY (line 7)</li> </ul> <pre>CREATE TABLE INSTRUCTOR(     InstructorID VARCHAR(5),     FirstName VARCHAR(15),     LastName VARCHAR(15),     DateOfBirth DATE,     Level INTEGER,     PRIMARY KEY (InstructorID) );</pre>	<b>3</b>
7(d)	<p><b>1 mark</b> per bullet point</p> <ul style="list-style-type: none"> <li>• Alter table student</li> <li>• Add an appropriate identifier with suitable data type</li> </ul> <pre>ALTER TABLE STUDENT ADD TelNum VARCHAR;</pre>	<b>2</b>
7(e)	<p><b>1 mark</b> per bullet point</p> <ul style="list-style-type: none"> <li>• Select lesson date and lesson time</li> <li>• From table LESSON</li> <li>• Where InstructorID = "Ins01"</li> <li>• And lesson date is greater than today's date</li> </ul> <pre>SELECT LessonDate, LessonTime FROM LESSON WHERE InstructorID = "Ins01" AND LessonDate &gt; #####;</pre>	<b>4</b>

Question	Answer	Marks
2(a)	<p><b>1 mark</b> per bullet point to <b>max 4</b>  <b>Max 3</b> if all generic, descriptions not related to benefits to the surgery  <b>Max 3</b> for a list with no expansions</p> <ul style="list-style-type: none"> <li>• Linked tables can be set up</li> <li>• ...the staff in the surgery can set up tables for the pets and their owners and link them by common attributes</li> <li>• To reduce / eliminate data redundancy</li> <li>• ...the staff in the surgery usually only needs to enter data once // in the file system data is probably repeated unnecessarily in different files</li> <li>• Improved data integrity</li> <li>• ... e.g. if they are searching for an owner's pets then all results for the owner should be returned</li> <li>• Privacy is improved</li> <li>• ... e.g. different views can be given to different users in the surgery. E.g. the receptionists cannot see the pet's medical notes</li> <li>• Referential integrity can be enforced // Unwanted or accidental deletion of linked data is prevented</li> <li>• ... e.g. the staff in the surgery cannot accidentally delete an owner's record while there are pets belonging to that owner // Staff cannot enter an appointment for a pet that does not exist</li> <li>• Program-data dependence is overcome</li> <li>• ...e.g. the staff in the surgery can add another attribute to the pet table without affecting the data already stored or the queries already written</li> <li>• More complex searches and queries can be executed ... e.g. the staff in the surgery can set up a query to only return the names of pets who have not been seen for over a year</li> </ul>	<b>4</b>
2(b)(i)	<p><b>1 mark</b> per bullet point to <b>max 1</b></p> <ul style="list-style-type: none"> <li>• OwnerFirstName, OwnerLastName and TelephoneNumber are repeated for owners with more than one pet.</li> <li>• OwnerFirstName, OwnerLastName and TelephoneNumber are not dependent on the primary key of the PET table.</li> </ul>	<b>1</b>

Question	Answer	Marks
2(b)(ii)	<b>1 mark per bullet point</b> <ul style="list-style-type: none"> <li>PET (<u>PetID</u>, PetName, PetBreed, PetDateOfBirth)</li> <li>OWNER (<u>OwnerID</u>, OwnerFirstName, OwnerLastName, TelephoneNumber)</li> <li>A linking table between PET and OWNER</li> <li>Composite primary key made up of the primary keys of the other two tables and no extra attributes in the linking table, for example, PET_OWNER (<u>PetID</u>, <u>OwnerID</u>)</li> </ul>	<b>4</b>
2(c)(i)	Structured Query Language // SQL	<b>1</b>
2(c)(ii)	<b>1 mark per bullet point</b> <ul style="list-style-type: none"> <li>ALTER TABLE APPOINTMENT</li> <li>ADD PRIMARY KEY (AppointmentID); //</li> <li>ADD UNIQUE (AppointmentID);</li> </ul>	<b>2</b>
2(c)(iii)	<b>1 mark for each correct line</b> <ul style="list-style-type: none"> <li>SELECT Time, PetID (FROM APPOINTMENT)</li> <li>WHERE StaffID = "JK1" AND Date = "02/02/2021"</li> <li>ORDER BY Time DESC;</li> </ul>	<b>3</b>
2(d)(i)	<b>1 mark for each bullet point to max 2 × 2</b> <ul style="list-style-type: none"> <li>Double entry // The data from the form is entered twice (by two different people)</li> <li><u>and</u> <b>automatically</b> compared</li> <li>Visual check // the data is compared (by two different people) after entry</li> <li>... to the <u>paper</u> form <b>manually</b></li> </ul>	<b>4</b>
2(d)(ii)	<b>1 mark per validation to max 2</b>  For example: <ul style="list-style-type: none"> <li>Time can have range check to make sure it is within the opening hours of 09:00 and 16:50</li> <li>Date can have existence check to compare against list of dates they are open</li> </ul>	<b>2</b>
2(e)(i)	<b>1 mark for each correctly completed term</b>  The <b>client-server</b> model has one <b>server</b> that stores all the data for the surgery. The other computers are <b>clients</b> . When a user requests data, a request is sent to the <b>server</b> .	<b>4</b>

Question	Answer	Marks
2(e)(ii)	<p><b>1 mark per correct method to max 2</b></p> <ul style="list-style-type: none"> <li>• Usernames and Passwords</li> <li>• Biometrics // fingerprint recognition // iris scanner</li> <li>• Two-step verification</li> <li>• Token authentication // use of dongle // swipe cards</li> </ul>	<b>2</b>