1.3 Exchanging Data: Databases, Networks and Web Technologies: 30-minute Pack

OCR H446 · Calculators not allowed · about 30 minutes

Total Marks

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A college admissions website used by Grange Sixth Form College includes a form asking applicants to enter their predicted grade points, and checks that the value entered is between 1 and 100 before the form is submitted.

(a) Distinguish between client-side scripting and server-side scripting, giving one example language typically used for each. **(4)**

(b) Explain one advantage of performing the grade point validation (checking the value is between 1 and 100) client-side rather than server-side. **(3)**

(c) The website also checks an applicant's login password against the college's database. Explain why this check must be carried out server-side rather than client-side. **(2)**

(Total for Question 1 is 9 marks)

© Revision Library | revisionlibrary.co.uk

2 A sixth form college stores exam entry data in a single unnormalised table. An extract is shown below, where each student record contains a repeating group of module entries:

StudentID	StudentName	TutorGroup	{ModuleCode, ModuleName, Grade}
1001	Amara Okafor	12A	{(COMP101, Computer Systems, A), (COMP102, Programming, B)}
1002	Liam Fraser	12B	{(COMP101, Computer Systems, B)}

(a) State why the table above is not in first normal form (1NF). **(2)**

(b) Convert the table into 1NF. Show the resulting table, including its primary key. **(3)**

(c) The 1NF table from part (b) contains partial dependencies. Identify the partial dependencies present and produce a set of tables in second normal form (2NF), stating the key of each table. **(4)**

(d) Explain the purpose of converting a database to third normal form (3NF), and state whether any further transitive dependency needs to be removed from the tables produced in part (c). **(3)**

(Total for Question 2 is 12 marks)

© Revision Library | revisionlibrary.co.uk

3 A college database contains the following relational tables:
Student(StudentID, Name, TutorGroup)
Module(ModuleCode, ModuleName)
Enrolment(StudentID, ModuleCode, Grade)
StudentID is the primary key of Student, ModuleCode is the primary key of Module, and (StudentID, ModuleCode) is the composite primary key of Enrolment, with StudentID and ModuleCode as foreign keys.

(a) Write an SQL query to list the Name of every student enrolled on the module with ModuleCode 'COMP101' who achieved a Grade of 'A'. **(3)**

(b) Write an SQL query to display each TutorGroup along with the number of students in that tutor group. **(3)**

(c) Write an SQL statement to add a new column called DateOfBirth, of type DATE, to the Student table. **(2)**

(Total for Question 3 is 8 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.3 Exchanging Data: Databases, Networks and Web Technologies: 30-minute Pack

Question 1

(a) Client-side scripting: code is executed in the user's browser/on the user's device (oe) **B1**

(a) Correct example client-side language, e.g. JavaScript **B1**

(a) Server-side scripting: code is executed on the web server before the page/response is sent to the client (oe) **B1**

(a) Correct example server-side language, e.g. PHP, Python, or ASP.NET **B1**

(a) Answer: Client-side: runs in the browser (e.g. JavaScript). Server-side: runs on the web server before the page is sent (e.g. PHP).

(b) States that the data does not need to be sent to the server and a response waited for before the error is shown (oe) **M1**

(b) Explains this reduces the load on the server and the amount of data sent over the network, since invalid entries are never transmitted (oe) **A1**

(b) Explains this gives the applicant immediate/faster feedback, improving the user experience (oe) **A1**

(b) Answer: Client-side validation checks the value in the browser immediately, so invalid entries are never sent to the server, reducing server load and network traffic and giving the applicant instant feedback.

(c) Client-side code can be viewed and modified by the user (e.g. via browser developer tools), so it cannot be trusted to enforce a security check (oe) **B1**

(c) The database holding the passwords is only accessible from the server, so the comparison must be performed there (oe) **B1**

(c) Answer: Client-side code is visible and editable by the user, so it cannot be trusted to enforce security, and only the server has access to the password database, so the check must happen there.

Question 2

(a) States that the table contains a repeating group / multi-valued attribute (oe) **B1**

(a) Identifies the specific repeating group in context, i.e. a student can have more than one set of ModuleCode/ModuleName/Grade values in one record **B1**

(a) Answer: It is not in 1NF because each student record can contain a repeating group of module data (more than one ModuleCode/ModuleName/Grade set per StudentID).

(b) Removes the repeating group so that each student-module combination becomes a separate row **M1**

(b) Correct attributes shown for every row: StudentID, StudentName, TutorGroup, ModuleCode, ModuleName, Grade **A1**

(b) Correctly identifies the composite primary key as (StudentID, ModuleCode) **A1**

(b) Answer:

StudentID	StudentName	TutorGroup	ModuleCode	M
1001	Amara Okafor	12A	COMP101	C S
1001	Amara Okafor	12A	COMP102	P
1002	Liam Fraser	12B	COMP101	C S

Primary key: (StudentID, ModuleCode)

(c) Identifies that StudentName and TutorGroup depend only on StudentID (part of the composite key), not on the whole key **M1**

(c) Identifies that ModuleName depends only on ModuleCode (part of the composite key), not on the whole key **M1**

(c) Correctly splits into Student(StudentID, StudentName, TutorGroup) and Module(ModuleCode, ModuleName) with appropriate primary keys **A1**

(c) Correctly retains Enrolment(StudentID, ModuleCode, Grade) with a composite primary key (StudentID, ModuleCode) and both attributes as foreign keys referencing Student and Module **A1**

(c) Answer: Student(StudentID [PK], StudentName, TutorGroup); Module(ModuleCode [PK], ModuleName); Enrolment(StudentID [PK, FK], ModuleCode [PK, FK], Grade)

(d) Defines a transitive dependency as a non-key attribute depending on another non-key attribute, rather than directly on the primary key (oe) **M1**

(d) Explains that removing transitive dependencies reduces data redundancy and prevents update/insertion/deletion anomalies (oe) **A1**

(d) Correctly judges, with justification, that no further transitive dependency exists in the part (c) tables, since every non-key attribute (StudentName, TutorGroup, ModuleName, Grade) depends directly on its table's primary key **A1**

(d) Answer: 3NF removes transitive dependencies to reduce redundancy and avoid anomalies. In this case the part (c) tables are already in 3NF, as no non-key attribute depends on another non-key attribute.

© Revision Library | revisionlibrary.co.uk

Question 3

(a) Correct SELECT and FROM clauses referencing Student and Enrolment (or an equivalent JOIN) **M1**

(a) Correct join condition linking Student.StudentID = Enrolment.StudentID **M1**

(a) Correct WHERE clause filtering ModuleCode = 'COMP101' AND Grade = 'A', cao **A1**

(a) Answer: SELECT Name FROM Student, Enrolment WHERE Student.StudentID = Enrolment.StudentID AND Enrolment.ModuleCode = 'COMP101' AND Enrolment.Grade = 'A';

(b) Correct SELECT of TutorGroup and a COUNT aggregate function on StudentID (or *) **M1**

(b) Correct GROUP BY TutorGroup clause **A1**

(b) Fully correct, syntactically valid statement cao **A1**

(b) Answer: SELECT TutorGroup, COUNT(StudentID) FROM Student GROUP BY TutorGroup;

(c) Correct ALTER TABLE Student keyword and table name **M1**

(c) Correct ADD COLUMN syntax with column name and DATE data type, cao **A1**

(c) Answer: ALTER TABLE Student ADD DateOfBirth DATE;

© Revision Library | revisionlibrary.co.uk