**RELATIONAL DATABASE FINAL EXAMS SOLUTIONS**

**QUESTION 1**

**Answer all parts**

Central Repairs is a new motor repairs facility set up this year. It needs to store data on all cars and owners that pass through its premises as well as details of all mechanics that work there. It also has to keep records on work carried out that may have been issued to each vehicle. The repairs facility wishes to store the following data about the Owners, Owner ID, Owner name, address, and contact number. For each mechanic, it wishes to store ID number, name, and year of qualification, address and contact number. For each car it requires the following information to be stored its registration number colour, engine capacity, and manufacturer. For each Repair, it requires to store Repair ID, Fixed/Not Fixed, and Cause of fault.

When the car is repaired, an invoice of work completed is assigned to the repair. Thus only one repair can generate one invoice and each invoice is generated by one repair. If the car cannot be repaired, no invoice will be created. For each Invoice, it stores Invoice ID, Invoice Date, Invoice Details and Amount. Each invoice needs to be signed off by an employee. There are a number of mechanics working on a particular car repair and each mechanic could work on one or more than one cars each day.

1. List the entities present in the above specification.
2. List the relationships that exist between these entities?

1. Draw an Entity Relationship (ER) diagram to represent the above scenario.
2. Develop a relational schema for your ER diagram.

**Answer (a), (b) and (c)**

The table below shows the results from a software testing laboratory. An *experiment* can be carried out on a piece of *software* multiple times and an experiment is always assigned a single *scientist*.

1. Propose a valid primary key for the table in its current form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *software* | *experiment* | *result* | *scientist* | *completionTime* |
| 1 | 1 | pass | Marco | 06/11/2017 17:20:21 |
| 1 | 2 | pass | Marco | 06/11/2017 17:30:40 |
| 1 | 3 | pass | Marco | 06/11/2017 17:40:02 |
| 2 | 1 | pass | Paulo | 06/13/2017 12:30:00 |
| 2 | 2 | fail | Paulo | 06/15/2017 11:20:01 |
| 3 | 1 | fail | Marco | 06/21/2017 09:20:30 |

(b) How would you reduce redundancy in the schema? Include your reasoning in your answer.

(c) Show your restructured schema including data.

**QUESTION 3**

**Answer all parts**

1. Taking into account, union / intersection between two relations, for instance relation R and S, what conditions must be fulfilled before one can have R ∩ S or R U S.
2. Which of these (union, intersection and difference) relational algebra operators are associate and commutative?

1. What is Duplicate elimination in the context of Projection?
2. Assume the following relation EMPLOYEE has the following tuples:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Office | Department | Rank |
| Smith | 400 | CS | Assistant |
| Jones | 220 | Econ | Adjunct |
| Green | 160 | Econ | Assistant |
| Brown | 420 | CS | Associate |
| Smith | 500 | Fin | Associate |

Write the relational algebra expressions to

1. Select employees in the CS department.

1. Select only those Employees with last name Smith and who are Assistant Professors.
2. Select only those Employees who are either Assistant Professors or work in the Economics department.

1. List the names and departments of all employees.

(v) Show the names of all employees working in the CS department.

**QUESTION 4**

**Answer (a), (b) and (c)**

Consider the following relational schema for a training college:

**student**(studentID, firstName, lastName, DoB, major, level)

**class**(classID, meetsAt, roomNo, *teacherID*)

**enrolled**(*studentID*, *classID*)

**teacher**(teacherID, firstName, lastName, *departmentID*)

**department**(departmentID, name, campus)

The primary keys are underlined. The foreign keys are highlighted in blue colour. A student is uniquely identified by a student ID. Classes are uniquely identified by their ID. Students are enrolled in classes (see enrolled table). Classes are taught by teachers. One teacher can teach many classes and each teacher is affiliated with a certain department.

Write SQL queries to answer the following:

1. Write statements to create all the tables from relational schema described above. Select appropriate data types for each attribute. Remember to set the primary and foreign keys.
2. List the IDs and full names of students on level 4 whose major is “Business” and are enrolled in more than two classes.
3. List the names of all the teachers who are working in the “Computing Department” located in “Dublin” campus.

**QUESTION 5**

**Answer all parts**

Consider the following relational schema for a company:

**country**(countryID, countryName, *RegionID*)

**employee**(employeeID, firstName, lastName, email, phoneNo, hireDate, salary, *jobID*, *managerID*, *departmentID*)

**job**(jobID, jobTitle, minSalary, maxSalary)

**locations**(locationID, address, *countryID*)

**department**(depatmentID, departmentName, *managerID*, *locationID*)

**region**(regionID, regionName)

The primary keys are underlined. The foreign keys are highlighted in italics. Each region has many countries, while each country has many (company) locations. Each location has many departments and each department has many employees working in it. Many employees can work under each job title.

Write SQL queries to answer the following:

1. Write a query to display the full name and hire date for all employees who were hired in year 1998.
2. Write a query to get the number of employees with the same job.
3. Write a query to get the average salary for all departments employing more than 10 employees.
4. Write a query to find the full name and salary of the employees who earn more than 50000 and work in any of the IT departments.

1. Write a query to get the department name and number of employees in the department.

**QUESTION 6**

**Answer all parts**

1. What is an Object Oriented Database? How is it different from Relational Databases?
2. What is the definition of distribution transparency in the context of Distributed Database Systems? Define Concurrency and Failure Transparencies.
3. List three advantages and three disadvantages of the Distributed Database Systems.

**QUESTION 1**

**Answer all parts**

The following narrative describes an IT training office’s course schedules:

The IT training office would like to record all data on the IT courses they deliver and the teachers who deliver the courses. Each teacher could give many courses and each course could be delivered by many teachers. Data stored on teachers are employee number, name, address, area of speciality and date hired. For each course, they store a course Id, course title and the award level. This level can be 4, 5, 6, 7 or 8. No teacher would deliver the same course twice.

Each course can be given three to four times a year. Thus each time a course is delivered, a new course schedule is created. Each course schedule has a date, time and duration. The duration is recorded as the number of months the course will be delivered for. This can range from six to 24 months. The location ID where the course would be delivered is also stored. Each course schedule is delivered by only one teacher. Each **course** would have only one location associated with it regardless of who is teaching it or when it is taught. Other data stored on location are the location name, address and room size of room. This room size is described as small, medium or large.

1. List the entities present in the above specification.

1. List the relationships that exist between these entities?
2. Draw an Entity Relationship (ER) diagram to represent the above scenario.
3. Convert your ERD to a set of relations.

|  |  |  |  |
| --- | --- | --- | --- |
| *id* | *Name* | *Ingredients* | *Notes* |
| 1 | Tea | Tea leaves,  Water | Green tea is best |
| 2 | Coffee | Coffee beans,  Water | Arabica is best |
| 3 | Ham Sandwich | Ham,  Bread,  Butter,  Lettuce | You can add mayo |
| 4 | Milkshake | Milk,  Ice-cream | You can add flavourings |
| 5 | Tomato soup | Chopped tomatoes,  Olive oil,  Garlic,  Onion | Cream is optional |

(a) What form is the above table?

(b) Normalise the table to 1NF. Explain what you did and show the resulting normalised schema.

© What data types did you choose for each attribute in your normalised schema?

(d) Is there redundancy in your 1NF schema? Propose how you would remove this redundancy.

**Answer all parts**

Assume the following relations:

**People**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| StaffID | Name | Age | WorkPlaceID | HomeTownPcode | Gender |
| 1 | Jennie | 30 | 102 | 7345 | Female |
| 2 | Tommy | 22 | 105 | 7388 | Male |
| 3 | Van | 31 | 101 | 7365 | Male |
| 4 | Noel | 38 | 101 | 7345 | Male |
| 5 | Sandy | 23 | 103 | 7355 | Sandy |

**Company**

|  |  |
| --- | --- |
| WorkPlaceID | CompanyName |
| 101 | SAP |
| 102 | Microsoft |
| 103 | Google |
| 104 | Oracle |
| 105 | Apple Inc. |

**HomeTown**

|  |  |
| --- | --- |
| HomeTownPcode | NameCity |
| 7335 | New York |
| 7345 | Dublin |
| 7355 | Boston |
| 7365 | Berlin |
| 7388 | Paris |

Given the following relations called Project, Mapping, Employee and Department, construct the relations that will result from the following relational algebra queries and outline the details:

1. πcompanyName(Company)

1. πName(σAge>25(People))

1. σStaffID>2∨Age=31(People)

1. πName, CompanyName (σPeople.WorkPlaceID=Company.WorkPlaceID (People X Company))

1. How many rows will be there if one takes the Cartesian product of HomeTown and Company? Illustrate the details that will be displayed.
2. Write down the equivalent SQL statements for relational expressions shown in 1 (a), 1(b) and 1(c).

**QUESTION 4**

**Answer all parts**

1. Define the following terms by illustrating them with a working example.
   1. Data Definition Language
   2. Data Manipulation Language

Presented is a brief representation of a database for a Delivery company:

CUSTOMER (CUSTOMER-No, NAME, ADDRESS, CITY, STATE, CODE, DISCOUNT)

ORDER (ORDER-No, ORDER-DATE, ARRIVAL-DATE, CUSTOMER-NO)

PRODUCT (PRODUCT-No, DESCRIPTION, FINISH, UNIT-PRICE, QTY-ON-HAND)

REQUEST (ORDER-No, PRODUCT-No, QTY-ORDERED)

The primary keys are underlined

Write SQL commands to perform the following operations:

1. List the Customer Numbers and Customer Names of all customers living in Dublin City
2. Select the quantity of the products ordered by a particular product Number.

1. Update the Unit price of all products by 5%.

1. Insert the following information into the Order Table (22, 12/09/2017, 22/09/2017, 12)

**QUESTION 5**

**Answer all parts**

The following is the description of a database table for a property for rent

Property(propertyNumber, Property\_Name, Address, No\_of\_rooms, Rent\_per\_month, Property\_Type)

1. Write an SQL UPDATE statement to update the rent in all properties by €100 which has a property type of detached.
2. Write an SQL UPDATE statement to update the number of rooms for the property number 10 to have 8 rooms (i.e. as if it has been renovated to include two more rooms).
3. Write an SQL DELETE statement which deletes all records from the Property table which have less than four rooms.

1. Write statements to create the table from relational schema described above. Select appropriate data types for each attribute. Remember to set the primary key.

1. Write a SQL statement to select all the properties that are located in Cork.

**QUESTION 6**

**Answer all parts**

(a) Discuss each of the following concepts in the context of relational data models?

(i) Attribute

1. Tuple
2. Degree
3. Cardinality
4. By using a diagram illustrate the ANSI-SPARC three-level architecture. Compare and contrast the three levels of this model.
5. List and explain three advantages and three disadvantages of the Database Management Systems.

**SECTION A – TWO QUESTIONS TO BE ATTEMPTED**

**QUESTION 1**

**Answer all parts**

A football club’s database contains the following table called “Player”. Each player is assigned to one or more team. The assignment is done through the “TeamID” column.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PlayerID** | **FName** | **LName** | **Tel** | **Position** | **DoB** | **TeamID** | **Coach** |
| P101 | Tom | Lee | 448841 | Forward | 31/5/01 | U18 | Pete |
| P101 | Tom | Lee | 448841 | Forward | 31/5/01 | U16 | Nick |
| P276 | Jeff | O’Neill | 568761 | Goalie | 6/12/00 | U18 | Pete |
| P276 | Jeff | O’Neill | 568761 | Goalie | 6/12/00 | U21 | Colin |
| P860 | Ken | Smith | 855110 | Midfield | 2/4/99 | U21 | Colin |
| P112 | Steve | Murphy | 348873 | Forward | 26/3/94 | Senior | Mark |

1. (i) Define the term *deletion anomaly.*

(ii) Consider the table above, and describe in words, with specific reference to the table, a situation where a deletion anomaly could occur.

(b) (i) Define the term *update anomaly.*

(ii) Consider the table above, and describe in words, with specific reference to the table, a situation where an update anomaly could occur.

(c) For this section, assume that the primary key for the above table is {PlayerID,TeamID}:

(i) Define the term *insertion anomaly*.

(ii) Give an example of a situation where an insertion anomaly could occur and explain why this would happen.

(d) Define the following terms:

(i) *Irreducible* *functional dependency*.

(ii) *Inconsistency.*

**QUESTION 2**

**Answer all parts**

1. What are the different levels of Database Architecture? Explain each level.
2. List five functions of a Database Management System (DBMS)? Briefly explain each function you have listed.
3. A database has been created for a concert promoter. This database records details about concerts and the artists who perform in them. The database stores the following information; {concert\_ID, venue\_name, date, director\_ID, director\_name, artist\_ID, artist\_name, artist\_no\_songs}.

* Every concert has a unique identifier (concert\_ID), which can be used to identify the venue\_name, date, director\_ID and director\_name of a concert. Each concert has one concert director who is responsible for managing the concert.
* Each concert director has a unique id (director\_ID) and his/her name is also recorded.
* Similarly, each artist has a unique id (artist\_id) from which we can find his/her name.
* Many artists can perform at a given concert and an artist can perform at many concerts.
* Each artist performs a fixed number of songs at a given concert (artist\_no\_songs).

Suppose we have the following relation that stores all the information:

Concert (**Concert\_ID**, Venue\_name, Date, Director\_ID, Director\_name, **Artist\_ID**, Artist\_name, Artist\_no\_songs)

1. Explain the criteria for a table to be classified as 1NF. State whether the given table structure is in 1NF and give a reason for your answer.
2. Explain the criteria for a table to be classified as 2NF and state if the given table structure is already in 2NF. If the given table is not in 2NF, decompose it so that it is in 2NF.
3. Explain the criteria for a table to be classified as 3NF and state if table(s) you have after part (ii) are already in 3NF. If the table(s) are not already in 3NF, decompose them so that they are.

An airline company needs a new database to be developed to store all the necessary information about flights it runs, the passengers that travel on those flights and the aircraft that are used. The following is further detail.

* Each regular **scheduled flight** has a unique flight number. This can be used to identify the departure airport, destination airport, and scheduled departure time of a flight. There can be many instances of a particular scheduled flight – we can consider these to be individual “*flights*”. Each instance will have a unique date.
* Each **passenger** can be identified by his/her passport number, from which we can find the passenger’s name, address and date of birth.
* Each **aircraft** has a unique registration number, from which we can find the make and model of the aircraft, and also the capacity (number of seats).
* When a passenger books a **flight**, we need to record the seat number to which that passenger is assigned. We also need to record the date of the flight.
* A passenger can take many flights with the airline.
* An aircraft can be assigned to many different flights and a scheduled flight can be carried out by many different aircraft, i.e. different planes on different dates.

1. Draw an entity relationship diagram for above specification. Provide only the final version of ER diagram. List attributes for each entity type in the diagram. Document assumptions you made when drawing the ER diagram.
2. Convert (map) the ER diagram to relational schema.

**QUESTION 4**

**Answer all parts**

Consider the following relational schema:

**Hotel** (hotelNo, hotelName, city)

**Room** (roomNo, *hotelNo*, type, price)

**Booking** (hotelNo, *guestNo*, dateFrom, dateTo, *roomNo*)

**Guest** (guestNo, guestName, guestAddress)

The primary keys are underlined. Write SQL queries to answer the following:

1. Write CREATE TABLE statement for each table in the relational schema described above. Select appropriate data types for each attribute. Remember to set the primary and foreign keys.
2. What is the average price of a room within a hotel whose hotelNo=131?
3. List guests with their name and address who have booked a room from “13/01/2016” to “19/01/2016”.
4. List total number of hotels in the database that have less than 10 rooms.

**QUESTION 5**

**Answer all parts**

Consider the following relational schema of a database for storing information about a construction company:

**WORKER** (WORKER-ID, WORKER-NAME, HOURLY-RATE, SKILL-TYPE, *SUPERVISOR-ID*)

**BUILDING** (BLDG-ID, BLDG-ADDRESS, BLDG-TYPE, QUALITY-LEVEL, STATUS)

**ASSIGNMENT** (*WORKER-ID, BLDG-ID*, START-DATE, NUM-DAYS)

Write SQL queries to answer the following:

1. Delete workers whose skill type is ENGINEER.
2. Find buildings and their start date with status = “INCOMPLETE”.
3. List the start date of buildings whose quality level is “TOP”.
4. List the average HOURLY-RATE of the workers who are working in a building with BLDG-ID= “11”.
5. List the worker details who has the highest hourly rate within the company.

**QUESTION 6**

**Answer all parts**

1. What is the difference between Entity Integrity and Referential Integrity?
2. Identify and discuss the 4 main types of transparency in a DDBMS.
3. List and discuss 5 disadvantages of DDBMS.

**QUESTION 1**

A university needs a new database to be developed to store all the necessary information about faculties, courses, modules, lecturers, and students. The following is further detail:

1. A faculty is described by its name and locations.
2. A course is described by its courseID, name, and duration (in number of semesters).
3. A module is recorded using its moduleID, title, credits, and duration (in contact hours).
4. Each lecturer has a lecturerID, first name, last name, email, phone and up to four qualifications.
5. Each student is described by a studentID, first name, last name, date of birth, age, email, address, phone, and grade point average.

The university contains a number of faculties. Each faculty can offer any number of courses and each faculty can be located in more than one geographical location. A course consists of many modules and each module belongs to only one course. Many lecturers work in a faculty, but a lecturer can only work in one faculty. Each faculty has a faculty head, but a lecturer can be head of only one faculty. Each lecturer can teach any number of modules, and a module can be taught by more than one lecturer. A student can enrol for any number of courses and each course can have any number of students. When a student completes a module, the student’s grade and the completion dates are recorded.

1. Draw an Entity Relationship Diagram (ERD) for the above set of requirements. Provide only the final version of the ERD diagram. Document any assumptions that you have made when drawing the diagram. List the attributes for each entity type in the diagram.
2. Map the above designed Entity Relationship Diagram into the relational schema.

**UESTION 2**

A Veterinary practice database contains the following table called “PetConsultation”. Each visit of a pet to the vet is recorded in a separate table, which is linked to using the “VisitID” field.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Pet ID** | **Pet Name** | **Species** | **DoB** | **Owner** | **Phone** | **VisitID** | **Vet** | **VetPhone** |
| P001 | Rex | Dog | 21/07/10 | Jane Eyre | 8769877 | T657 | Joe | 23747423 |
| P001 | Rex | Dog | 21/07/10 | Jane Eyre | 8769877 | T809 | Ken | 74747323 |
| P002 | Misty | Cat | 13/05/11 | Rick Rockett | 9870987 | T598 | Susan | 43444442 |
| P002 | Misty | Cat | 13/05/11 | Rick Rockett | 9870987 | T786 | Joe | 23747423 |
| P003 | Harvey | Dog | 08/04/02 | Jane Eyre | 8769877 | T392 | Liz | 23362233 |
| P004 | Elvis | Cat | 09/01/07 | Nora O’Dell | 2283932 | T765 | Susan | 43444442 |

1. (i) Define the term *update anomaly*.

(ii) By making reference to the above table, describe a situation where an update anomaly could occur.

1. (i) Define the term *deletion anomaly*.

(ii) By making a reference to the above table, describe a situation where a deletion anomaly could occur.

(c) For this section, assume that the primary key for the above table is {PetID, VisitId, Vet}.

1. Define the term *insertion anomaly*.
2. Give an example of a situation where an insertion anomaly could occur and explain why this would happen.

(d) Define the following terms:

1. Inconsistency.
2. Logical data independence.
3. Functional dependency.
4. Fan trap.
5. Partial key.

**QUESTION 3**

1. Describe how the Cartesian product operator of relational algebra functions as applied to two tables A and B such that A X B. Comment on the size and shape of the resulting relation.
2. Explain why only using the Cartesian product to join two tables is useless. What other operator is needed to make Cartesian product useful? Explain why this operator is needed.
3. Given the following relations called Project, Mapping, Employee and Department, construct the relations that will result from the following relational algebra queries:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| project code | project title | project manager | project budget | Project Table |  |  | |
|  |  |  |  |  |  |  | |
| PC010 | Pensions System | M Phillips | 24500 |  |  |  | |
| PC045 | Salaries System | H Martin | 17400 |  |  |  | |
|  |  |  |  |  |  |  | |
| project code | Employee # | hourly rate | Mapping Table | Employee # | Employee name | Department # |
|  |  |  |  |  |  |  |
| PC010 | S10001 | 22.00 |  | S10001 | A Smith | L004 |
| PC010 | S10030 | 18.50 |  | S10030 | L Jones | L023 |
| PC010 | S21010 | 21.00 |  | S21010 | P Lewis | L004 |
| PC045 | S10010 | 21.75 |  | S10010 | B Jones | L004 |
| PC045 | S10001 | 18.00 |  |  |  |  | |
|  |  |  |  |  |  |  | |
| Department # | Department Name | Department Table |  |  |  |  | |
|  |  |  |  |  |  |  | |
| L004 | IT |  |  |  |  |  | |
| L023 | Pensions |  |  |  |  |  | |

1. Π Employee#, Employee Name, Department Name (σ Employee.Department# = Department.Department# (Employee X Department))
2. σ Mapping.Employee# = Employee.Employee#  ProjectCode = PC010 (Mapping X Employee)
3. Π EmployeeName (σ Mapping.Employee# = Employee.Employee#  Project.ProjectCode = Mapping.ProjectCode  project manager = H Martin (Project X Mapping X Department))

**QUESTION 4**

Consider the following relational schema for a junior football club:

**Player** (player\_id, fname, lname, birth\_date, address, phone)

**Plays\_On** (player\_id, team\_id, position)

**Team** (team\_id, team\_name, team\_level, coach\_id)

**Coach** (coach\_id, fname, lname, phone, email, coach\_type)

The primary keys are underlined.

A player is uniquely identified by a player id. Each team is uniquely identified by a team id. Each player can play on more than one team and each team can have many players who are on the panel for that team. Each team has one coach and a coach can work with more than one team.

Note: player\_id and team\_id in the Plays\_on table are foreign keys to player\_id in the Player table and team\_id in the Team table, respectively.

Write SQL queries to answer the following:

1. Create **Player**, **Plays\_On**, **Team**, and **Coach** tables from the relational schema described above. Select appropriate data types for each attribute. Remember to set the primary and foreign keys.
2. List the team\_name of all teams who play at senior level (team\_level = “Senior”).
3. List the personal details and playing position of all players who are on team U15A (team\_name = “U15A”).
4. For each team, print the number of players who belong to that team.

Consider the following relational schema for a music database:

Artist(ArtistID, Name, StartDate, RetireDate)

Album(AlbumID, Name, ReleaseDate, ArtistID)

Track(TrackID, Name, Length, AlbumID)

RecordLabel(LabelID, Name)

Album-RecordLabel(AlbumID, LabelID)

The primary keys are underlined. An artist is completely identified by an ArtistID, an album is completely identified by an AlbumID, a track is completely identified by a TrackID, and a record label is completely identified by a LabelID. An album can only have one artist. A track can only belong to one album. An album can be produced by more than one record label and a record label can produce many albums.

Write SQL queries to answer the following:

1. Write queries to construct the Album and Track tables. You must include primary keys and foreign keys use appropriate datatypes.
2. Write a query to pick out the artists who started after the 1st of January of the year 2000.
3. Write a query to pick out the track information for the band Metallica’s album called “Reload”.
4. Write a query that will pick out the names of all albums by the Foo Fighters.

1. Write a query that will pick out the number of record labels that were involved in the production of each album.
2. Describe the roles of the Concurrency, Backup and Recovery services in an RDBMS?
3. Distributed databases are used to store large amounts of data that cannot be managed in a single database. Identify and discuss five disadvantages of this approach.
4. Describe how a deductive database works. Define all components of a deductive database.

**QUESTION 1**

Consider the following table called “Employee”. Each employee works on projects that are mentioned in “ProjectAssigned” column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **Address** | **ProjectAssigned** |
| Emp101 | Sean | Duffy | 21 lower street, city center | Project001 |
| Emp101 | Sean | Duffy | 21 lower street, city center | Project008 |
| Emp119 | Hannah | Murphy | 03 upper street, city west | Project002 |
| Emp119 | Hannah | Murphy | 03 upper street, city west | Project008 |
| Emp710 | Olga | Carter | 77 upper street, city center | Project003 |

1. (i) Define the term *deletion anomaly.*
2. Consider the table above, and describe in words, with specific reference to the table, a situation where a deletion anomaly could occur.

(b) (i) Define the term *update anomaly.*.

(ii) Consider the table above, and describe in words, with specific reference to the table, a situation where an update anomaly could occur.

(c) For this section, assume that the primary key for the above table is {EmployeeID and ProjectAssigned}.

(i) Define the term *insertion anomaly*.

(ii) Give an example of a situation where an insertion anomaly could occur and explain why this would happen.

(d) Define the following terms:

(i) *Functional dependency*.

(ii) *Transitive functional dependency.*

1. What are the different components of a Database Management System (DBMS)? Explain each component.
2. What are the three phases of a database design methodology? Explain each phase.
3. A database has been created that records details about teachers (and the modules they teach) in a college. The database stores the following information; TeacherID, Fname, Lname, Telephone, Email, ModuleCode, ModuleName, ModuleDuration.

* Every teacher has a unique identifier (TeacherID), which can be used to identify the Fname, Lname, Telephone, and Email of a teacher.
* Each module has a unique id (ModuleCode) so that we can get module details e.g. name of the module and its duration (recorded in weeks).
* Each teacher teaches more than 1 module.

Suppose we have the following relation created that stores all the information:

TeacherModule(TeacherID, Fname, Lname, Telephone, Email, ModuleCode, ModuleName, ModuleDuration)

1. Explain the criteria for a table to be classified as 1NF. State whether the given table structure is in 1NF and give a reason for your answer.
2. Explain the criteria for a table to be classified as 2NF and state if the given table structure is already in 2NF. If the given table is not in 2NF, decompose it so that it is in 2NF.
3. Explain the criteria for a table to be classified as 3NF and state if table(s) you have after part (ii) are already in 3NF. If the table(s) are not already in 3NF, decompose them so that they are.

**QUESTION 3**

A music company needs a new database to be developed to store all the necessary information about musicians, albums, songs, and live performances. Following is further detail.

* Each musician in the company has a unique SSN, Fname, Lname, an address, and a phone number.
* Each album recorded has an albumID, title and an author (musician).
* Each album has a number of songs on it. Each song has a unique SongID, name, length, and a track number.
* Each song is performed by one or more musicians, and a musician may perform a number of songs.
* Musicians also perform in live performances. The company wants to keep track of the location and time of each performance that a given musician took part in.

1. Draw an entity relationship diagram for above specification. Provide only the final version of ER diagram. List attributes for each entity type in the diagram. Document assumptions you made when drawing the ER diagram.
2. Convert (map) the ER diagram to relational schema.

**SECTION B – TWO QUESTIONS TO BE ATTEMPTED**

**QUESTION 4**

Consider the following relational schema, in which the primary keys have been underlined:

**Student** (snum, fname, sname, major, level, age)

**Class** (cname, meets\_at, room, fid)

**Enrolled** (snum, cname)

**Faculty** (fid, fname, sname, deptid)

A student is uniquely identified by a student number snum, and information about their name, major, grade level, and age is stored. Classes are uniquely identified by their name, and their meeting time, room number, and faculty instructor are stored. Enrolled lists which students (snum) are enrolled in which classes (cname). Faculty is uniquely identified by fid, and information about their name and department is stored.

Write SQL queries to answer the following:

1. Write SQL to create **Student**, **Class**, **Faculty**, and **Enrolled** tables from the relational schema provided above. Select appropriate data types for each attribute. Also remember to set primary and forgien keys.

(b) Find fname, sname, and major of all students who are seniors (level = ‘SR’).

**©** List the detail of faculty members who taught only in room JS003.

1. For each level, print the level and total number of students belong to that level.

Consider the following relational schema, in which the primary keys have been underlined:

**Customer** (cust\_id, fname, lname)

**Employee** (emp\_id, fname, lname, manager\_emp\_id)

**Cust\_Order** (order\_id, cust\_id, emp\_id, sale\_price)

A customer is uniquely identified by a customer id, and information about their name is stored. An employee is uniquely identified by employee id, and information about their name and manager id is stored. Cust\_Order table stores information regarding orders, customers who place orders, employees who deal with orders and sale price of orders.

Write SQL queries to answer the following:

1. Write SQL to alter the **Employee** table by changing *fname* and *lname* attributes to *first\_name* and *last\_name* respectively.

Display the total sales price of all orders from each customer.

(c) Display the total sales price of all orders from each customer where only customers who spent more than 1500 (in total) are considered.

1. Update the fname and lname of customer with cust\_id = 120 to “Andy” “Carter”.
2. Delete all orders that have a sale\_price less than 50.
3. Explain the difference between Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Lanaguage (DCL).
4. Cardinality of relationships has three types; one-to-one, one-to-many, and many-to-many. Explain and give an example of each of the cardinality type using appropriate diagrams.
5. Describe the potential advantages of distributed databases.

**QUESTION 1**

**Answer all parts**

Below is a table from a database that contains information about software developers. A developer is assigned to a single room and assigned a team lead. Each developer was asked to provide their personal mobile phone numbers for use as a contact phone number.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **devID** | **Name** | **mobile** | **dob** | **room** | **teamLead** |
| 0 | A Carmac | 555-1111 | 1992-2-14 | RD2 | M Stewart |
| 1 | M Gates | 555-3333 | 1991-1-3 | RD2 | M Stewart |
| 2 | K Woz | 555-4444 | 1988-11-12 | RD2 | M Stewart |
| 3 | K Gambale | 555-9999 | 1989-11-29 | 5 | J Taggart |
| 4 | H Codd | 555-7777 | 1979-1-11 | RD5 | M Vans |
| 5 | K Enfield | 555-6666 | 1982-7-12 | 1 | J Taggart |

1. Are there any consequences when deleting the row with the devID value of 4? What would this undesirable side-effect be called?
2. What is an update anomaly? Provide an example of such an anomaly using the table above.
3. Identify two functional dependencies from the table above based *only* on the information in the table. Try and use the correct notation (X → Y) where X and Y are sets of attributes from the above table.
4. Suggest suitable data types for the above attributes based on the above information. What does the primary key “devID” look like in terms of key type and data type?

**QUESTION 2**

**Answer all parts**

The following information represents the structure of a database from a baking website:

Each recipe has a recipeID, the name of the recipe, instructions, difficulty to make on a scale from 1 to 5, preparation time in minutes, and cooking time in minutes.

Each recipe can have one or more ingredients. Each ingredient has an ingredientID, ingredient name, and unit of measurement (“g” or “ml” etc.).

Each guest has a guestID, name of the guest, and a password. Each guest can review multiple recipes but only review each recipe once.

A review contains, a guestID, a recipeID, the text of the review, and a rating from 1 to 5.

1. Identify and list all the entities present in the above specification.
2. Identify and list all the relationships present in the above specification.
3. Draw an entity-relationship diagram for the above specification.
4. Create a relational schema from your entity-relationship model.
5. Identify and explain all the keys and the types of keys involved in relational schema from answer.

**QUESTION 3**

**Answer all parts**

1. Explain the cardinalities (or degrees) of relationships in relational database design. Provide simple examples in the form of a sentence and the cardinality of the relationship that is represented in the sentence.

**(5 marks)**

The following table contains the information from a car hire company.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chassisNo | model | engineSize | colour | mileage |
| C3B26CDC-D732-24BB-997C-CFAB0D424FCD | puma | 1.9 | indigo | 57737 |
| CFFE6F6D-A75A-D1F6-7F58-8385801AEA49 | puma | 1.9 | violet | 54828 |
| 0230A5A4-AB09-8023-0041-D3F195FD3134 | espace | 1.4 | green | 68526 |
| 24C61FC3-3732-C40E-1549-B52D3EEA449E | c4 | 1.6 | violet | 19920 |
| 52FA82BC-A85F-62B5-AD5D-667ACFE870D2 | mondeo | 2 | yellow | 75673 |
| 4F54FF8C-09AA-EA7A-3F19-E36AB25D5581 | espace | 1.4 | violet | 20702 |
| 03EC2711-8A7F-A41D-BD8D-0627D64D5CD8 | c4 | 1.6 | green | 58937 |
| 0A9E52E8-9A1B-0E26-B960-7002C6245CB8 | mondeo | 2 | indigo | 134105 |
| BFABE077-5A51-9706-DC4F-8EF6FB6C9337 | puma | 1.9 | red | 120259 |
| 8425DE5D-AA95-F047-F4D8-7D0A04AF2EBC | espace | 1.4 | red | 29251 |

1. In terms of database normalisation, identify the form of this table? Explain your answer.
2. Based on the data in the table, explain how you would restructure the table into 3NF. Mention any interim stages of normalisation if they exist. The normalised data is not required, only the normalised structures.

**QUESTION 4**

**Answer all parts**

Consider the following relational schema for a company:

**Region** (RegionCode, RegionName)

**Customer** (CustCode, CustName, CustAdress, CustBalance, RegionCode)

**EmployeeType** (EmployeeType, HourlyPayRate,)

**Employee** (EmployeeNo, EmployeeName, EmployeeAddress, RegionCode, EmployeeType)

**Product** (ProductCode, ProductName, UnitPrice, StockOnHand)

**Order** (OrderNo, CustCode, ProductCode, QtyOrdered, EmployeeNo)

The primary keys are underlined.

A Region is uniquely identified by a RegionCode. Each Customer is uniquely identified by a CustCode. Each Customer can order more than one Product and each Product can have many Customers. Each Order is dealt with by an employee and an Employee can work with more than one Order.

Write SQL queries to answer the following:

1. Create **Customer**, **Product**, and **Order** tables from the relational schema described above. Select appropriate data types for each attribute. Remember to set the primary and foreign keys.
2. List the Employee numbers, names and address of all the employees working in the Region whose code is MUN and the name is John Smith.
3. List the Employee names and address of all the employees working in the EmployeeType whose code is 1 and the hourly rate of pay is €40.

1. Write the statement to update the unit price of ProductCode S106 by €2.

**QUESTION 5**

**Answer all parts**

Consider the following relational schema for a music database:

Author(AuthorID, AuthorName, Address, TelephoneNo, PublisherCode)

Book (BookID, Name, ReleaseDate, Price, AuthorID)

Publisher(PublisherID, Name, Address, AuthorID)

The primary keys are underlined. An Author is completely identified by an AuthorID, a book is completely identified by an BookID, a Publisher is completely identified by a PublisherID. A book can only have one Author. A book can only belong to one Publisher.

Write SQL queries to answer the following:

1. Write queries to show all the book information of all Books in the Book table which includes the name Harry Potter. Sort Descending in Price.
2. Write a query to pick out and show all the details of Authors who released books last year.
3. Write a query to show the number of authors who have written a book.
4. Write a query that will illustrate the books and their various Publishers. Group by PublisherID.
5. Write a query to delete the Telephone Number column from the Author table.
6. Explain the difference between logical and physical data independence.
7. What is a Deductive Database? Describe how it works.
8. A Distributed Database Management System (DDBMS) has many characteristics. Outline and discuss 5 of these characteristics.

**QUESTION 1**

1. Define the following terms in relation to Entity Relationship Modelling
   1. Entity Type
   2. Recursive Relationship
   3. Attribute
   4. Derived Attribute
   5. Weak Entity type

**Read the following which describes the data requirements of a company. The company consists of several departments and have asked you to construct an ER Diagram**

A Department employs a Worker. A department may employ many workers but a worker can only work for (and must work for) one department. Details stored about the worker include worker ID(unique), Name and Address. Details stored about Departments include Department No (unique), Name and Location.

A Worker participates in a Project. A Worker may participate in many projects and a project must have many workers assigned to it. Details stored about Project include Project ID (unique) and Project Name.

A Department controls a Project. A department may control none or many projects but a project must be controlled by only one department. A project consumes Raw Materials which need to be ordered. A Project may contain many raw materials and a raw material may be consumed in Many Projects. Details stored about Raw Materials include Material ID (Unique), Description and Stock. The amount of a raw material (quantity) consumed on a particular project must also be stored.

An Order for Raw Materials may contain many raw materials and a raw material may appear on many orders. Details about the Order include Order No (Unique), Order Date and Status. The amount of a raw material ordered on a particular order as well as the unit price of a raw material on the same order must also be stored.

A supplier supplies an Order. An order must have only one supplier but a supplier may have many orders. Details about Supplier include Supplier ID (unique), Supplier Name, Street, Town, County, Contact No.

1. Draw an Entity Relationship Diagram (ERD) for the above set of requirements. Document any assumptions that you made when drawing the diagram.
2. List the attributes for each entity type and any constraints in the diagram.
3. Map the above designed Entity Relationship diagram into a set of relations.

**QUESTION 2**

1. List the components of a DBMS? Explain how each type of component works.
2. A database has been created for a Concert Ticket Booking System that records details about customers and their Ticket Bookings. The database stores the following information; CustCode, Fname, Lname, Telephone, Payment Type, Balance, BookingNo, Booking Type, Booking Date, Paid.

Every customer has a unique identifier (CustCode), which can be used to identify the Fname, Lname, Telephone, Payment Type and Balance of a customer.

Each Booking has a unique identifier (BookingNo) that is used to identify the booking details e.g. date of the booking and the type of booking.

Each customer can make more than 1 booking.

Suppose we have the following relation created that stores all the information:

CustCode, Fname, Lname, Telephone, Payment Type, Balance, BookingNo, Booking Type, Booking Date, Paid.

1. When can a table to be classified as 1NF. State whether the above table structure is in 1NF and provide a reason for your answer.
2. When can a table to be classified as 2NF and state if the above table structure is already in 2NF. If the given table is not in 2NF, decompose it so that it is in 2NF.

1. When can a table to be classified as 3NF and state if table(s) you have after part (ii) are already in 3NF. If the table(s) are not already in 3NF, decompose them so that they are in 3NF.

The following tables form part of a database held in a relational DBMS:-

Rental Company (CompanyNo, CompanyName, city)

Car (CarReg, CompanyNo, type, make, price)

Booking (CompanyNo, CustomerNo, dateFrom, dateTo, CarReg)

Customer (CustomerNo, CustomerName, CustomerAddress)

Describe the relations that would be produced by the following relational algebra operations:

* 1. ΠRental CompanyNo (price  150 (Car) )

* 1. Rental Company.Rental CompanyNo  Car.Rental CompanyNo(Rental Company  Car)
  2. ΠRental CompanyName (Rental Company Rental Company.Rental CompanyNo  Car.Rental CompanyNo (price  150 (Car)) )
  3. Customer (dateTo ≥ ‘1-Jan-2015’ (Booking))
  4. Rental Company Rental Company.Rental CompanyNo  Car.Rental CompanyNo (price  150 (Car)) )

1. Write relational Algebra operations to carry out the following
2. List all the Rental companies
3. List the Customer Numbers, Customer Names and Customer Address for all customers
4. List the details of all cars and the customers that booked them.

**QUESTION 4**

Consider the following relations:

**Supplier** (SupplierID, SupplierName, Address)

**Part** (PartID, PartName, Color)

**Catalog** (*SupplierID, PartID*, Cost)

The Supplier relation stores information about the supplier. The Part relation store information about the parts. The Catalog relation lists the prices charged for parts by Suppliers. The primary keys are underlined.

Write SQL queries to answer the following:

1. Write SQL to create **Supplier**, **Part**, and **Catalog** tables from the relational schema provided above. Select appropriate data types for each attribute. Also remember to set primary and foreign keys.

1. Find the “part names” of parts supplied by Acme Widget Suppliers and no one else.
2. Find the names of suppliers who supply only red parts.
3. For every supplier that supplies a green part, print the name and price of the most expensive part that she supplies.

Consider the following relations:

**Flight** (flight-no, from-city, to-city, distance, depart-time, arrive-time, price)

**Aircraft** (aircraft-id, aircraft-name, cruisingrange)

**Certified** (*employee-id, aircraft-id*)

**Employee** (employee-id, employee-name, salary)

The relations keep track of airline flright information. The Employee relation describes pilots and other kind of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly. The primary keys in each relation have been underlined.

Write each of the following queries in SQL.

1. For each pilot who is certified for more than three aircraft, find the employee-id and the aircraft-name for which she or he is certified.
2. Print the name and salary of every non-pilot employee.

1. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles.
2. Find the names of pilots certified for aircraft named “Boeing 747”.
3. A customer wants to travel from Madison to New York. List the choice of departure times from Madison if the customer wants to arrive in New York before 18:00 (6 pm).
4. Explain and provide examples for each of the following types of relationships:
   1. Binary Relationship
   2. Ternary Relationship
   3. Unary Relationship
5. Describe the following terms:
   1. Entity integrity
   2. Referential integrity

1. List the advantages and disadvantages of Distributed Database Management Systems.

**QUESTION 1**

Consider the following table called “Student”. Each student registers to a number of modules that are mentioned in “ModuleTaken” field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StudentID** | **FirstName** | **LastName** | **ModuleTaken** | **ModuleName** |
| 02S001 | Humberto | Luke | WEB001 | Web Development |
| 02S001 | Humberto | Luke | OOP001 | Object Oriented Programming I |
| 02S001 | Humberto | Luke | OOP002 | Object Oriented Programming II |
| 02S002 | Michael | O’grady | WEB001 | Web Development |
| 02S002 | Michael | O’grady | OOP001 | Object Oriented Programming I |
| 02S003 | Floyd | Painter | DB001 | Introduction to Databases |
| 02S004 | Kay | Rayonman | OOP001 | Object Oriented Programming I |
| 02S004 | Kay | Rayonman | WEB001 | Web Development |

1. Define the following terms:
2. *Deletion anomaly:*  if you try to delete on piece of information you might delete other information that you do not want to delete.
3. *Update anomaly:* If you try to update something with the same information, you might update other lines of the table. The same above.
4. *Insertion anomaly:* occurs when certain attributes cannot be inserted into the database without the presence of other attributes. For example this is the converse of delete anomaly we can't add a new course unless we have at least one student enrolled on the course. StudentID. ModuleTaken.

(b) Answer the following questions:

1. By making a reference to the above table, describe a situation where an update anomaly may occur.

If I try to update the Module taken for Floyed Painter to WEB001, it affects all the rest of the students registered in the same module.

1. By making a reference to the above table, describe a situation where a deletion anomaly may occur.

If I try to use the delete command to using the Module taken as reference to remove the student registered on WEB001, it will also delete all the others students registered in the same module.

The primary key for the above table is {*StudentID* and *ModuleTaken*}. By making a reference to the above table, describe a situation where an insertion anomaly may occur.

We can't add a new course unless we have at least one student enrolled on the Module taken. StudentID. ModuleTaken. (without primary key)

(c) Define the following terms:

1. *Referential Integrity.* Primary Key in one table is referencing a foreign key in another table for consistency.
2. *Determinant.: (*Primary key – functionally dependent on it ) The determinant of a functional dependency refers to the attribute or group of attributes. Determinants should have the minimal number of attributes necessary to maintain the functional dependency with attribute (s) on the right hand side.

A

B is functionally dependent on A

1. What are the different types of database users? Explain each type by providing examples.

End-user

1. Define the following algebraic operators:
2. Selection
3. Projection
4. Cartesian Product
5. Union
6. Intersection
7. A database has been created that records details about customers and their orders. The database stores the following information; CustomerID, Fname, Lname, Telephone, OrderID, OrderDate, OrderPrice.

* Every customer has a unique identifier (CustomerID), which can be used to identify the Fname, Lname, Telephone of a customer.
* Each order has a unique identifier (OrderID) that is used to identify order details e.g. date of the order and its price.
* Each customer can make more than 1 orders.

Suppose we have the following relation created that stores all the information:

CustomerOrder(CustomerID, Fname, Lname, Telephone, OrderID, OrderDate, OrderPrice)

1. When can a table to be classified as 1NF. State whether the above table structure is in 1NF and provide a reason for your answer.

CustomerOrder(CustomerID, Fname, Lname, Telephone,)

Order\_Date ( CustomerID, OrderID, OrderDate, OrderPrice)

1. When can a table to be classified as 2NF and state if the above table structure is already in 2NF. If the given table is not in 2NF, decompose it so that it is in 2NF.

CustomerOrder(CustomerID, Fname, Lname, Telephone,)

Order\_Date ( CustomerID, OrderID, OrderDate)

Order (CustomerID, OrderID, OrderPrice)

1. When can a table to be classified as 3NF and state if table(s) you have after part (ii) are already in 3NF. If the table(s) are not already in 3NF, decompose them so that they are in 3NF.

CustomerOrder(CustomerID-PK, Fname, Lname, Telephone,)

Order\_Date ( CustomerID- PK, OrderID - FK, OrderDate)

Order (OrderID -PK, OrderPrice)

A database is to be designed for a Car Rental company. The information required includes a description of cars, customers, staff, and garage. Following is further detail.

* Each car is described by its make model, year of production, engine size, fuel type, number of passengers, registration number, and rent price.
* Each customer is recorded using customerID, fname, lname, phone, email, and address.
* Each staff has staffID, fname, lname, phone, and address.
* Each garage is described by garageID, name, and address.

Customers rent cars and the staff work in garage. Each customer can rent many cars and each car is rented by many customers. Each garage has a number of cars to be rented. Many staff members work in a garage. Each time a customer rents a car the hire date of the car and return date are recorded.

(a) Draw an Entity Relationship Diagram (ERD) for the above set of requirements. Provide only the final version of ERD diagram. Document any assumptions that you made when drawing the diagram. List attributes for each entity type in the diagram.

(b) Map the above designed Entity Relationship diagram into relational schema.

Consider the following relational schema, in which the primary keys have been underlined:

**Employee** (emp\_id, fname, sname, birth\_date, address, phone, manager\_id)

**Works\_On** (emp\_id, project\_id, hours)

**Project** (project\_id, project\_name, department\_number)

An employee is uniquely identified by an employee id. Each project is uniquely identified by a project id. Each employee works on many projects and each project can have many employees working on it. Note: emp\_id and project\_id in Works\_On table are foreign keys to emp\_id in employees and project\_id in project table, respectively. Manager\_id is a foreign key to emp\_id in employee table.

Write SQL queries to answer the following:

1. Create **Employee**, **Works\_On**, and **Project** tables from the relational schema provided above. Select appropriate data types for each attribute. Remember to set primary and foreign keys.
2. Find employees’ fname, sname and phone who have same managers.
3. Find employees’ fname and sname who work on projects for more than 5 hours.
4. Write SQL to alter the **Works\_On** table created above to include another field called *days* (this will store number of days an employee work on a project).

**QUESTION 5**

The following set of relations describes a sales system, in which the primary keys have been underlined:

**Region** (RegionId, RegionName)

**Manager** (ManagerId, ManagerName, ManagerAddress, Salary)

**Branch** (BranchId, BranchName, Location, RegionId, ManagerId)

**Product** (ProductId, Description, UnitPrice, VatRage)

**BranchStock** (BranchId, ProductId, StockOnHand, MinStockLevel)

**Customer** (CustCode, CustName, CustAddress)

**Sale** (SaleNo, CustCode, BranchId, ProductID, QtySold)

Write SQL queries to answer the following:

1. List the locations of all branches in the region with Region Id “Munster”.
2. List the BranchId and BranchName for the Branch managed by John Smith.
3. The branch with branchId “L123” is closing down. Delete all the sales records for this branch.
4. The company is organizing a business lunch for customers and managers. Prepare one list that includes both managers and clients, showing their names and addresses.
5. Update address of a manager with ManagerId “m008” to “Dublin 8, Dublin, Ireland”.

**QUESTION 6**

1. Write down and explain shortly the problems we may face when storing information in traditional flat file based systems.
2. With regard to Codd’s 12 Rules, explain each of the following:

Guaranteed Access Rule

Physical Data Independence

Logical Data Independence

Integrity Independence

Systematic Treatment of NULL Values

1. Describe shortly the differences between distributed and centeralized databases.

The table below is an example of a database table called “Sales”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Phone** | **Email** | **ProductName** | **ProductColour** | **Qty** |
| Bob Hope | 1234456 | bh@comp.org | Jeans | Black | 1 |
| Bob Hope | 1234456 | bh@comp.org | Hat | Blue Spots | 2 |
| Mary Murphy | 6548732 | m22@social.net | Hat | Red Spots | 8 |
| Mary Murphy | 6548732 | m22@social.net | Dress | Orange | 1 |
| Claire Toban | 3214632 | ct@tra.com | Hat | Red Spots | 3 |

1. (i) Define the term *deletion anomaly.*

(ii) Consider the table above, and describe in words, with specific reference to the table, a situation where a deletion anomaly could occur.

1. Write the SQL that would perform the deletion you specified in (ii) above.

(b) (i) Define the term *update anomaly.*.

1. Consider the table above, and describe in words, with specific reference to the table, a situation where an update anomaly could occur.
2. Write the SQL you would use to perform the update you stated in part (ii), that would result in an inconsistency.
3. For this section, assume that the primary key for the above table is {Name,ProductName,ProductColour}.

(i) Define the term *insertion anomaly*.

(ii) Give an example of a situation where an insertion anomaly could occur and explain why this would happen.

(iii) Write the SQL code you would use to perform the action stated in part (ii).

The following schema describes a database to store details about employees in a company. *(e.g. what department they are in and what project they are assigned to.)* Note that an employee is assigned to a project and works for a department, a department oversees a project. The employee may be on a project different to the one their department controls.

Department(dept\_id, dept\_name, dept\_location)

Project(project\_id, project\_name, *dept\_id*)

Employees(SSN, employee\_name, *dept\_id*, *project\_id*)

1. What is the role of the Project.dept\_id field in the Project table?
2. What effect would the following referential constraint (if used as part of a CREATE TABLE statement) have on the allowable values of the Employees.project\_id field?

FOREIGN KEY (project\_id) REFERENCES Project(project\_id)

1. The above set of relations contains three foreign keys (Project.dept\_id, Employees.dept\_id, and Employees.project\_id). If each is defined using ON DELETE CASCADE, explain what the effect of deleting an entry from the Course table would be. For full marks, you should consider its effect (if any) on all tables.

The following is an example of a database schema for storing information about students in a university:

Course(course\_id, course\_name)

Module(module\_code, module\_name, stage, *course\_id*)

Student(student\_id, student\_name)

Studies(*student\_id*, *module\_code*, grade)

**Rule 1**: A course can not be removed from the database if someone has, at some point, taken it.

**Rule 2**: If a module\_code changes, this should be kept consistent everywhere.

1. Declare any ON DELETE referential constraint(s) that are required to satisfy the above business rules.
2. Taking the following referential constraints.
   * ON UPDATE NO ACTION
   * ON DELETE SET NULL

Explain the difference between the two. *You must use example tables with sample-data to get full marks.*

Declare any ON UPDATE referential constraint(s) that are required to satisfy the above business rules.

(iii) Write the entire SQL CREATE TABLE statement for the Studies table, including the appropriate referential constraints and primary key (*your answer should be consistent with your answers for parts (i) and (ii)*).

The following example database table stores data relating to a general practitioner's patient list

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **patient\_name** | **next\_of\_kin** | **age\_group** | **illness** | **treatment** | **price** |
| Mary Doyle | Mark Doyle | 18-29 | Flu | Injection | 35 |
| John Clarke | Brian Clarke | 30-60 | Infection | Antibiotics | 40 |
| Peter Hoey | Claire Coyle | 30-60 | Fever | Antibiotics | 30 |
| Louise Murphy | Kate Murphy | 18-29 | Sprain | Bandaged | 20 |

The following functional dependencies exist:

1. fd1: {patient\_name} → {next\_of\_kin, age\_group,illness}
   * fd2: {age\_group, treatment} → {price}
2. fd3: {age\_group, illness} → {treatment}

(a) Based on the above provided functional dependencies:

(i) Using any of the functional dependencies supplied above, identify one error in the data and explain why this must be an error.

* 1. Define what is meant by a *partial key dependency* and identify where this occurs in the data provided.
  2. Write three business rules which correspond (in meaning) to each of the three provided functional dependencies (i.e. fd1, fd2 and fd3 listed above).

(b) What is meant by the term *non-loss decomposition?*

A database has been created that records details about animal patients in a vet hospital, including the animals names, their owner's details, and where they are kept during their stay. It contains the following data: {animal\_id, animal\_name, owner\_id, owner\_name, animal\_type, location}.

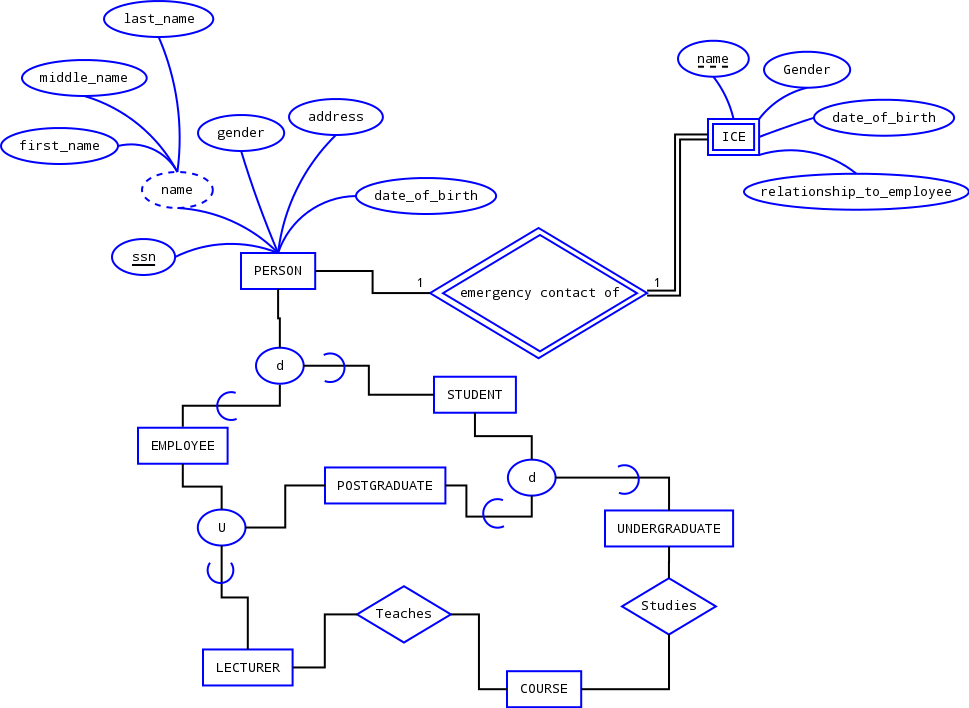
Every patient has a unique identifier, which can be used to identify the name of the animal and the owner

* Each owner has a unique id so that we can find their details
* Each type of animal is kept in a certain area, e.g. The cats in the cattery, the dogs in the kennels, the birds in the avery, etc.

The relation can be written as:

Patient(animal\_id, animal\_name, owner\_id, owner\_name, animal\_type, location)

1. Explain the criteria for a table to be classified as 1NF. State whether the given table structure is in 1NF and give a reason for your answer.
2. Explain the criteria for a table to be classified as 2NF and state if the given table structure is already in 2NF. If the given table is not in 2NF, decompose it so that it is in 2NF.
3. Explain the criteria for a table to be classified as 3NF and state if table(s) you have after part (ii) are already in 3NF. If the table(s) are not already in 3NF, decompose them so that they arE



Consider the ER Diagram below:

Explain the relationship between PERSON and ICE (In Case of Emergency)

Explain the *name* attribute of the PERSON entity

1. Again, with respect to the above diagram, describe the *specialisation* of the STUDENT entity.

(c)Assuming that each lecturer only teaches one class and that each class can only be taught by one lecturer:

(i) Name the relation cardinality ratio

(ii) What are the three possibilities for the relationship in terms or optionality?

(iii) Select one of the optionality possibilities from above and describe, with the aid of a diagram, the design decisions with respect to primary keys, foreign keys, amalgamation, etc.

**QUESTION 5**

(a) Convert each of the following relational algebra expressions into their equivalent SQL expressions:

(i) PROJECT gender, salary (employees)

(ii) PROJECT employee\_ssn (RESTRICT year\_joined > 2010 (employees)

* 1. managers RIGHT OUTER JOIN managers.ssn = employees.ssn employees

(b) With regard to the Three-Schema Architecture

(i) List the three levels.

(ii) Of the three levels listed above, choose one and discuss what it describes.

1. Name the process of transforming requests and results between levels.

(c)Data independence can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. In this context:

(i) Define *logical data independence.*

(ii) Describe *physical data independence.*

(iii) Describe, with the aid of an example, how a *view* can provide logical data independence.

**QUESTION 6**

1. There are those who criticise Relational Databases in favour of using Object Relational Databases instead. Outline and explain these criticisms and how an Object Relational Database addresses them.
2. Using an example, contrast how an Object Relational Database and a Relational Database can be used to represent data. Your answer should include:

(i) A sample scenario that could be represented either in an Object Relational Database or a Relational Database

(ii) A description (using a suitable Data Definition Language) of the data structures required to represent this scenario in both an Object Relational Database **and** a Relational Database.