

CS2312 Example Sheet 2

1. Consider the following relational schema for a database describing accommodation and activities available in national parks:

Park (<u>name</u> , region)	
Hotel (<u>id</u> , name, t-name)	t-name is a foreign key into Town
Town (<u>name</u> , p-name)	p-name is a foreign key into Park
Activity (<u>p-name</u> , <u>activity</u>)	p-name is a foreign key into Park

- i. Write an SQL to create a view called “Climbing” that contains the id, name and t-name of all Hotels situated in a park where “climbing” is an available activity.
 - ii. Write an SQL query that lists the name of each Town, with the name of the Park it is in and the number of hotels it contains. The list should be ordered first by Park name and then by Town name in ascending alphabetical order.
2. Consider the following relational schema for a database that contains patient details, including their ID, name, doctor and medical condition; doctors details, including their name, medical speciality and rank; and details of medical conditions, including the medical speciality appropriate to the treatment of that condition.

Patient(<u>ID</u> , name, doctorName, condition)
Doctor(<u>doctorName</u> , speciality, rank)
Condition(<u>condition</u> , speciality)

- i. Write SQL to define a view called “WrongDoc” that lists the ID, name, condition, doctorName and speciality for all those patients whose doctor has a speciality that is different from the speciality that is appropriate to the treatment of the patient’s condition.
- ii. Use the “WrongDoc” view you have created to write an SQL query that returns a duplicate free list of the name, speciality and rank of all doctors who are treating one or more patients with a condition for which the doctors speciality is different from the speciality that is appropriate to the treatment of the patient’s condition. The list should be in alphabetical order of doctor’s names (i.e., names beginning with A come before names beginning with B).

3. Consider the (IN)tegrated (S)chema for (U)niversity (L)ibr(AR)ies in Britain, detailed below:

•INSULAR Schema

- LIBRARY(libno,libname,location,rooms)
- BOOK(bookno,title,pages,authno (fk))
- AUTHOR(authno,authname)
- COPY(copyno,libno (fk),bookno (fk),cost)
- LOAN(copyno (fk), borrowerno (fk), duedate)
- BORROWER(borrowerno,name,age)

Most attribute names are self-explanatory. In the LIBRARY and BOOK tables, the rooms and pages attributes refer to the number of rooms in a library and to the number of pages in a book, respectively. The cost attribute in the COPY table refers to how much the library paid for the copy of a book. Attributes followed by (fk) are foreign keys, and underlined attributes are keys. Some features of the schema are:

- Each book has only one author.
- Each book has a distinct book number, known as ISBN, that is represented at the schema by the bookno attribute.
- It is important to differentiate between a BOOK and a COPY of a book. For instance, the book (Fundamentals of Database Systems, ISBN 0201542633) has several copies available within the same or different libraries (e.g., 3 copies at John Rylands, 1 at the CS dept. library, 2 at the UMIST library, 1 at Salford's library, etc.) The system assumes that each copy has a distinct copy number (copyno attribute) that uniquely identifies the copy independently of the library where it is situated.

- a) Provide SQL expressions that can answer the following questions:
- i. Retrieve the names and locations of libraries that have books written by the author whose name is "Codd".
 - ii. Retrieve the names of borrowers who have no books on loan.
 - iii. Retrieve the names of authors at least one of whose books is on loan.
 - iv. Retrieve the borrower's name and the number of books currently on loan for each borrower.
- b) Create an SQL view that contains the borrower name, duedate and copyno of all books borrowed by senior citizens (i.e., age > 65).
- c) Is the view above updateable? Why?