ASSIGNMENT 1

Q.1 List five responsibilities of a database-management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

Q.2 Explain the difference between two-tier and three-tier architectures. Which is better suited for Web applications? Why?

Q.3 Discuss the relative merits of procedural and nonprocedural languages.

Q.4 Consider the following University Schema :-

create table classroom

(building varchar(15),

room\_number varchar(7),

capacity numeric(4,0),

primary key (building, room\_number)

);

create table department

(dept\_name varchar(20),

building varchar(15),

budget numeric(12,2) check (budget > 0),

primary key (dept\_name)

);

create table course

(course\_id varchar(8),

title varchar(50),

dept\_name varchar(20),

credits numeric(2,0) check (credits > 0),

primary key (course\_id),

foreign key (dept\_name) references department

on delete set null

);

create table instructor

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

salary numeric(8,2) check (salary > 29000),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

create table section

(course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6)

check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),

year numeric(4,0) check (year > 1701 and year < 2100),

building varchar(15),

room\_number varchar(7),

time\_slot\_id varchar(4),

primary key (course\_id, sec\_id, semester, year),

foreign key (course\_id) references course

on delete cascade,

foreign key (building, room\_number) references classroom

on delete set null

);

create table teaches

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references instructor

on delete cascade

);

create table student

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

tot\_cred numeric(3,0) check (tot\_cred >= 0),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

create table takes

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

grade varchar(2),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references student

on delete cascade

);

create table advisor

(s\_ID varchar(5),

i\_ID varchar(5),

primary key (s\_ID),

foreign key (i\_ID) references instructor (ID)

on delete set null,

foreign key (s\_ID) references student (ID)

on delete cascade

);

create table time\_slot

(time\_slot\_id varchar(4),

day varchar(1),

start\_hr numeric(2) check (start\_hr >= 0 and start\_hr < 24),

start\_min numeric(2) check (start\_min >= 0 and start\_min < 60),

end\_hr numeric(2) check (end\_hr >= 0 and end\_hr < 24),

end\_min numeric(2) check (end\_min >= 0 and end\_min < 60),

primary key (time\_slot\_id, day, start\_hr, start\_min)

);

create table prereq

(course\_id varchar(8),

prereq\_id varchar(8),

primary key (course\_id, prereq\_id),

foreign key (course\_id) references course

on delete cascade,

foreign key (prereq\_id) references course

);

* 1. Find the names of all the students whose total credits are greater than 100
  2. Find the course id and grades of all courses taken by any student named 'Tanaka'
  3. Find the ID and name of instructors who have taught a course in the Comp. Sci. department, even if they are themselves not from the Comp. Sci. department. To test this query, make sure you add appropriate data, and include the corresponding insert statements along with your query.
  4. Find the courses which are offered in both 'Fall' and 'Spring' semester (not necessarily in the same year

1. Find the id and title of all courses which do not require any prerequisites.
2. Find the names of students who have not taken any biology dept courses
3. Write SQL update queries to perform the following :
   * 1. Give a 10% hike to all instructors
     2. Increase the tot\_creds of all students who have taken the course titled "Genetics" by the number of credits associated with that course.
     3. For all instructors who are advisors of at least 2 students, increase their salary by 50000.

Q.5 Explain division operation in Relational algebra with an example.

ASSIGNMENT 2

Create an ER diagram for each of the following enterprises

1. A railway system, which needs to model the following:
   1. Stations
   2. Tracks, connecting stations. You can assume for simplicity that only one track exists between any two stations. All the tracks put together form a graph.
   3. Trains, with an ID and a name
   4. Train schedules recording what time a train passes through each station on its route. You can assume for simplicity that each train reaches its destination on the same day, and that every train runs every day. Also for simplicity, assume that for each train, for each station on its route, you store (a) time in, (b) time out (same as time in if it does not stop), and (c) a sequence number so the stations in the route of a train can be ordered by sequence number.
   5. Passenger booking consisting of train, date, from-station, to-station, coach, seat and passenger name; for simplicity, don't bother to model passengers as entities.

Q.2 Design an ER diagram for keeping track of the scoring statistics of your favourite sports team. You should score the matches played, the scores in each match, the players in each match and individual player scoring statistics for each match. Summary statistics should be modelled as derived attributes with an explanation as to how they are computed.

Q.3 Consider the following schema: Suppliers (sid: integer, sname : string, address : string) Parts (pid : integer, pname : string, colour : string) Catalouge (sid : integer, pid : integer, cost : real). You can use either SQL or Relational Algebra

1. Find the name of suppliers who supply some red parts
2. Find the sids of suppliers who supply some red part or are at 221 packer

Q.4 An ER diagram can be viewed as a graph. What do the following mean in terms of the structure of an Enterprise schema?

1. The graph is disconnected.
2. The graph has a cycle

Assignment 3

Database Design and Normalization

This assignment has several parts. In the first part we work on basic manipulation of functional dependencies. Next, we use functional dependencies to normalize some toy relations into BCNF/3NF. And finally, we take a real life example and figure out the functional dependencies involved, and develop a normalized database design.

1. Functional dependencies: For each of the following sets of functional dependencies on a schema r(A, B, C, D, E)
   * Find a candidate key for this schema
   * Find the attribute closure of AB
   * AB --> C, D --> E, B --> E
   * A --> CD, B --> DE
   * AB --> C, C --> D
2. Normalization 1: For each of the above set of functional dependencies, decompose relation r into BCNF
3. Normalization 2: For each of the following sets of functional dependencies on a schema r(A, B, C, D, E),
   * Find the canonical cover by eliminating all extraneous attributes
   * Decompose relation r into 3NF based on the canonical cover
   * A --> CD, B --> DE, C --> D
   * A --> B, B --> C, A --> C, D --> E, B --> E, AD --> E

Real life example: Consider a database   
student(ID, name, courseID, year, semester, grade)  
instructor(ID, name, deptname, deptbudget)  
List the functional dependencies you would expect to hold on the above relations, and decompose them into BCNF.

ASSIGNMENT 4

Q.1 Suppose a well-balanced range-partitioning vector had been chosen for a relation, but the relation is subsequently updated, making the partitioning unbalanced. Even if virtual-processor partitioning is used, a particular virtual processor may end up with a very large number of tuples after the update, and repartitioning would then be required

A. Suppose a virtual processor has a significant excess of tuples (say, twice the average). Explain how repartitioning can be done by splitting the partition, thereby increasing the number of virtual processors.

B. If, instead of round-robin allocation of virtual processors, virtual partitions can be allocated to processors in an arbitrary fashion, with a mapping table tracking the allocation. If a particular node has excess load (compared to the others), explain how load can be balanced.

C. Assuming there are no updates, does query processing have to be stopped while repartitioning, or reallocation of virtual processors, is carried out? Explain your answer.

Q.2.

Consider the following extension to the tree-locking protocol, which allows both shared and exclusive locks:

A transaction can be either a read-only transaction, in which case it can request only shared locks, or an update transaction, in which case it can request only exclusive locks.

Each transaction must follow the rules of the tree protocol. Read-only transactions may lock any data item first, whereas update transactions must lock the root first.

Show that the protocol ensures serializability and deadlock freedom.

Q.3. In timestamp ordering, W-timestamp(Q) denotes the largest timestamp of any transaction that executed write(Q) successfully. Suppose that, instead, we defined it to be the timestamp of the most recent transaction to execute write(Q) successfully. Would this change in wording make any difference? Explain your answer.

Q.4 Devise a timestamp-based protocol that avoids the phantom phenomenon.

Q.5 Consider a variant of the tree protocol called the forest protocol. The database is organized as a forest of rooted trees. Each transaction Ti must follow the following rules:

The first lock in each tree may be on any data item.

The second, and all subsequent, locks in a tree may be requested only if the parent of the requested node is currently locked.

Data items may be unlocked at any time.

A data item may not be relocked by Ti after it has been unlocked by Ti .

Q.6

Show that the forest protocol does not ensure serializability.

Suppose that a B+-tree index on (dept name, building) is available on relation department. What would be the best way to handle the following selection?

(building < “Watson”) ∧ (budget < 55000) ∧ (dept name = “Music”)(depar tment)