DATABASE MANAGEMENT SYSTEMS

ETCS - 208

ASSIGNMENT - 1

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

MAIT

उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

Faculty : Dr. Anu Rathee

Submitted by : Khushi
Roll No : 03114813120
Semester : 4th
Branch : ITE
Group : ITE123

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

Ans (i) Interaction with the file manager
(ii) Integrity enforcement
(iii) Security enforcement
(iv) Backup and recovery
(v) Concurrency Control

If these responsibilities were not met by a database management system, the following problems can occur, respectively.

1. No DBMS can do without this, if there is no file manager interaction then nothing stored in the files can be retrieved.
2. Consistency constraints may not be satisfied, account balances could go below the minimum allowed, employees could earn too much overtime (e.g., hours > 80 hours) or, airline pilots may fly more than allowed by the law.
3. Unauthorized users may access the database, or users authorized to access part of the database may be able to access parts of the database for which they lack of authority. For, example, a high school student could get access to national defence secret codes, or employees could find out what their supervisiors earn.
4. Data could be lost permanently, rather than at least being avilable in a consistent state that existed prior to a failure.
5. Consistency constraints may be violated despite proper integrity enforcement in each transaction.
For example, incorrect bank balances might be reflected due to simultaneous withdrawals and deposits, an so on.

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

Ans

Two-Tier Database Architecture	Three-tier Database Architecture
1. It is a Client-Server Architecture.	It is a Web-based application.
2. In two-tier, the application logic is either buried inside the user interface on the client or within the database on the server (or both).	In three-tier, the application logic or process resides in the middle-tier, it is separated from the data and the user interface
3. Two tier architecture consists of two layers: Client Layer Database (Data Layer)	Three-tier architecture consists of three layers: Client Layer Buisness Layer Data Layer

4.	It is less secured as client can communicate with database directly.	It is secured as client is not allowed to communicate with database directly.
5.	Two-tier architecture runs slower.	Three-tier architecture runs faster.
6.	It is easy to build and maintain.	It is complex to build and maintain.
7.	It results in performance loss whenever the users increase rapidly.	It results in performance loss whenever the system is run on Internet but gives more performance than two-tier architecture.

Three tier architecture is best suited for Web Applications because :-
The chief benefit of three-tier architecture is that each tier runs on its own infrastructure, each tier can be developed simultaneously by a separate development team, and can be updated or scaled as needed without impacting the other tiers.

Other benefits are as follows:-
→ Faster development
→ Improved scalability
→ Improved reliability
→ Improved security

Ques 3 Discuss the relative merits of procedural and non-procedural languages.

Ans PROCEDURAL LANGUAGE:

In procedural languages, the program code is written as a sequence of instruction. User has to specify "WHAT TO DO" and also "HOW TO DO" (step by step procedure). These instructions are executed in the sequential order. These instructions are written to solve specific problems.

Examples of Procedural Languages:-

FORTAN, COBOL, ALGOL, BASIC, C and Pascal

NON-PROCEDURAL LANGUAGE

In the non-procedural languages, the user has to specify only "WHAT TO DO" and not "HOW TO DO".
It is also known as an applicative or functional language. It involves the development of the functions from other functions to construct more complex functions.

Examples of Non-Procedural Languages:-
SQL, PROLOG, LISP.

Ques 4 Explain division operation in Relational algebra with an example.

Ans The division operator of relation algebra, '$\div$' is defined as follows.
Let $r(R)$ and $s(S)$ be relations, and let $S \subseteq R$; that is, every attribute of schema S is also in schema R.
Then $r \div s$ is a relation on schema $R - S$. A tuple t is in $r \div s$ if and only if both of two conditions hold:

* t is in $\Pi_{R-S}(r)$
* For every tuple t_s in s, there is a tuple t_r in r satisfying both of the following:
  - $\rightarrow$ $t_r[S] = t_s[S]$
  - $\rightarrow$ $t_r[R-S] = t$

Ques 5 Write SQL queries

(a)
```
Select name
from student, course
where course.credits > 100;
```

(b)
```
Select course_id, grade
from takes natural join student
where student.name = 'Tanaka';
```

(c)
```
Select ID, name
from instructor, teaches
where teaches.course_id = 'CS-101' or teaches.course-id = 'CS-315' or
                                           teaches.course-id = 'CS-347';
```

```
d) (Select course-id
    from section
    where semester= 'Fall')
   intersect
   ( select course_id
     from section
      where semester = 'spring');
```

```
1.) Select course_id, title from course
    where course_id not in (select course_id from prereq);
```

```
2.) Select name from student
    where ID in (select ID from takes
                 where course_id in (select course_id from course
                                     where dept_name != 'Biology'));
```

```
3.)
 1) update instructor set salary = salary * 1.1;

 2) update student set tot_credit = tot_credit + (select credits from course where
                                                  course-title = 'Genetics');

         where ID in (select ID from student, course
                      where student.dept_name and course title = 'Genetics');
```

```
3)
  Update instructor set salary = salary + 5000
        where name in (select name from instructor, advisor
                       where instructor.ID = advisor.ID
                 group by instructor.name having count (s-ID) > 1);
```

DATABASE MANAGEMENT SYSTEMS

ETCS - 208

ASSIGNMENT - 2

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

MAIT

उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

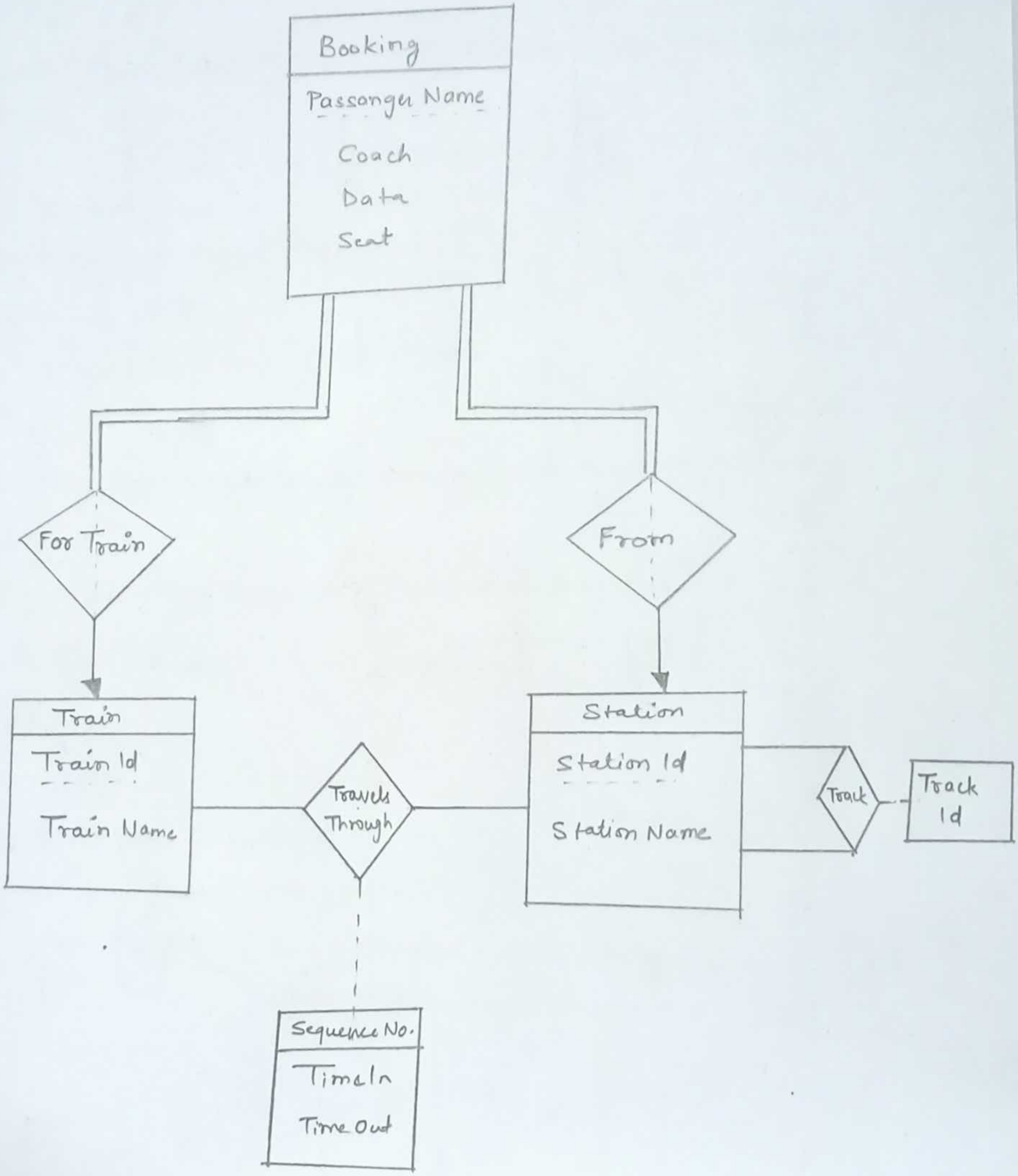
Faculty : Dr. Anu Rathee

Submitted by : Khushi
Roll No : 03114813120
Semester : 4th
Branch : ITE
Group : ITE123

Ques Create an ER diagram

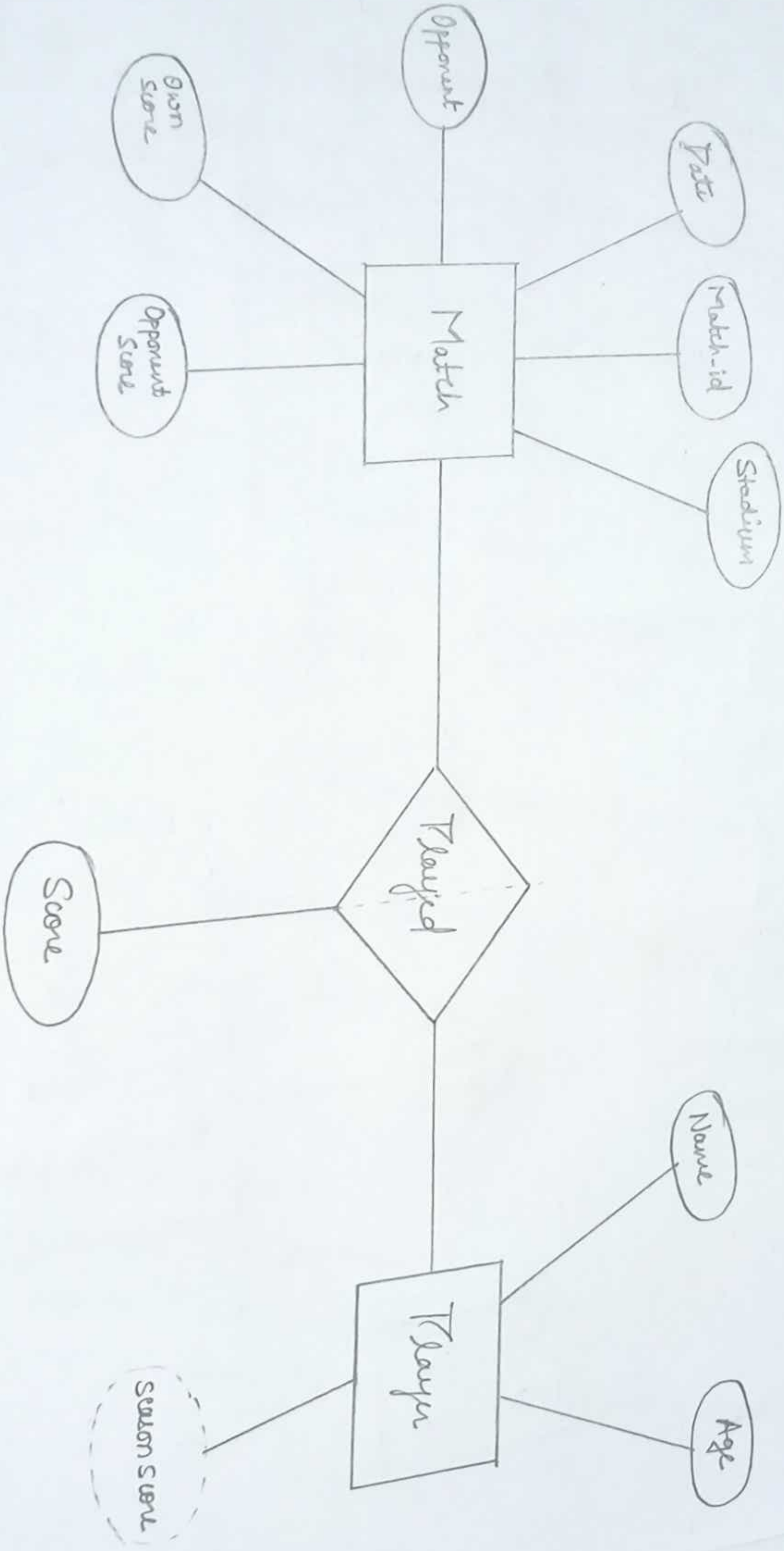
1) A railway station, which needs to model the follow :-

ER-diagram for Railway Station



Ques Design an ER diagram for keeping track of the exploits of your favourite sport team. You should store the matches played, the scores in each match, the player in each match and individual player statistics for each match. Summary statistic should be modeled as derived attribute.

Ans



E-R diagram for favourite team Statistics

Ques Consider the following schema:

Suppliers (sid: integer, sname: string, address: string)
Parts (pid: integer, pname: string, color: string)
Catalouge (sid: integer, pid: integer, cost: real).
You can use either SQL or Relational Algebra.

Ans (a) Find the name of the suppliers who supply some red parts.

Relational Algebra

$$\pi_{sname}\left(\pi_{sid}\left(\pi_{pid}\left(\sigma_{color='red'}\ Parts\right) \bowtie cataloge \bowtie\right) suppliers\right)$$

SQL

```
Select S.sname from
        Suppliers S, Parts P, Catalog C
  where P.color = 'red' and
        C.pid = P.pid and  C.sid = S.sid
```

(b) Find the sids of suppliers who supply some red parts or are at 221 packer.

Relational Algebra

$$\rho\left(R_1, \pi_{S.id}\left(\pi_{pid}\ \sigma_{color='red'}\ Parts\right) \bowtie catalog\right)$$

$$\rho\left(R_2\ \pi_{sid}\left(\sigma_{address='221\ packer'}\ Suppliers\right)\right.$$

$$R_1 \cup R_2$$

SQL

```
Select S.sid from Suppliers S
   where S.address = "221 packer"
      or S.sid In ( Select C.id from parts P, Catalog C
                     where P.color = 'red' and P.pid = C.pid)
```

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

(a) The graph is disconnected
If a pair of entity sets are connected by a path in an E-R diagram, the entity sets are related, through
A disconnected graph implies that there are pairs of entity sets that are unrelated to each other, hence may be better off to design separate databases (each corresponding to a connected subgraph).

(b) The graph has a cycle
If there is a cycle in the graph then every pair at entity sets on the cycle are related to eachother in atleast two distinct ways.

(c) The graph is acyclic
There is a unique path between every pair of entity sets and thus, a, unique relationship between every pair of entity sets.

(d) The ER diagram has the same entity set appearing several times.
it means we are missing some relationships in the model, leading to a bad design.

DATABASE MANAGEMENT SYSTEMS

ETCS - 208

ASSIGNMENT - 3

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

MAIT

उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

Faculty : Dr. Anu Rathee

Submitted by : Khushi
Roll No : 03114813120
Semester : 4th
Branch : ITE
Group : ITE123

Ques 1

1) AB → C , D → E , B → E

r (A, B, C, D, E)

Clearly, from the above diagram, we conclude that (ABD) is a candidate key
Now, attribute closure of AB ⇒ $(AB)^+$ = (ABCE)

Normalization

Prime attributes ⇒ A, B, D
Non - Prime attributes ⇒ C, E

The functional dependency AB → C, B → E are partial dependencies
Thus it is not in 2NF also.

Normalising

r (A B C)
r (D E)
r (B E)
r (A B D)

The above decomposition is clearly in BCNF because → A relation schema R is in BCNF if whenever a non-trivial functional dependency X → A holds in R, then X is superkey of R.

∴ Required Normalization ⇒ r (ABC), r (DE), r (BE) and r (ABD)

2) A → CD , B → DE

r (A, B, C, D, E)

Clearly, from the above diagram, we conclude that (AB) is a candidate key.
Now, attribute closure of AB ⇒ $(AB)^+$ = (ABCDE)

Normalization

Prime attribute → A, B
Non-Prime attribute → C, D, E

The functional dependencies A → CD and B → DE are partial dependencies.
Thus, it is not in 2NF also.

Normalising :

r (A C D)

r (B D E)

r (A B)

The above decomposition is clearly BCNF because → A relation schema R is in BCNF if ~~otherwise~~ whenever a non trivial functional dependency $X \to A$ holds in R, that X is superkey of R.

∴ Required normalization ⇒ r(ACD), r(B DE) and r(AB)

3) $AB \to C$, $C \to D$

r(A, B, C, D, E)

Clearly, from the above diagram, we conclude that (ABE) is a candidate key.

Now, attribute of colsure of AB ⇒ $(AB)^+ = (A\ B\ C\ D)$

Normalization

Prime attributes ⇒ A, B, E

Non Prime attributes ⇒ C, D

The functional dependency, $AB \to C$ is a partial dependency and $C \to D$ is a transitive dependency.

Thus it is not in 2NF also.

Normalising

Removing partial dependency ⇒

~~R (A B C)~~ r (A B C D)

~~R (C D)~~ r (A B E)

~~R (A B E)~~

~~The~~ Now, removing transitive dependency ⇒

r (A B C)

r (C D)

r (A B E)

The above decomposition is clearly in BCNF because → A relation schema R is in BCNF if whenever a non trivial functional dependency $X \to A$ holds in R, then X is a superkey of R.

∴ Required Normalization ⇒ r(A B C), r(C D) and r(A B E)

Ques 2

1) $A \to CD$, $B \to DE$, $C \to D$

Decomposing ⇒

$A \to C$ $\quad B \to D \quad C \to D$

$A \to D$ $\quad B \to E$

- For $A \to C$

$(A^+)_{\text{when } A\to C \text{ is present}} = ACD$

$(A^+)_{\text{when } A\to C \text{ is not present}} = ACD$

Since, we are getting same result so $A \to D$ is redundant, so eliminated.

- For $B \to D$

$(B^+)_{\text{when } B\to D \text{ is present}} = BDE$

$(B^+)_{\text{when } B\to D \text{ is not present}} = BE$

Since, we are getting different result so $B \to D$ is not redundant.

- For $B \to E$

$(B^+)_{\text{when } B\to E \text{ is present}} = BED$

$(B^+)_{\text{when } B\to E \text{ is not present}} = BD$

Since, we are getting different result so $B \to E$ is not redundant.

- For $C \to D$

$(C^+)_{\text{when } C\to D \text{ is present}} = CD$

$(C^+)_{\text{when } C\to D \text{ is not present}} = C$

Since, we are getting different result so $C \to D$ is not redundant.

∴ Required canonical cover

$A \to C \quad B \to D \quad C \to D \quad \Rightarrow \quad A \to C, \; B \to DE, \; C \to D$

$\quad\quad\quad B \to E$

Now, r(A B C D E)

Clearly, from the above diagram, we conclude that (AB) is a candidate key.

r (D, E)

r (A, D)

Prime attributes $\Rightarrow$ A, B

Non-Prime attributes $\Rightarrow$ C, D, E

The functional dependencies $A \to C$, $B \to DE$ are partial dependencies and
$C \to D$ is transitive dependency.

Thus, it not in 2NF also.

Normalising

Removing partial dependencies

r(A C D)

r(B D E)

r(A B)

Now, removing transitive dependencies

r(A C)

r(C D)

r(B D E)

r(A B)

The above decomposition is clearly is 3NF because → The relation schema R is third normal if, whenever a non trivial functional dependency $X \to A$ holds in R, either (a) X is a super key of R, or (b) A is a primary attribute of R.

$\therefore$ Required normalisation $\Rightarrow$ r(AC), r(CD), r(BDE), r(AB)

2) $A \to B$, $B \to C$, $A \to C$, $D \to E$, $B \to E$, $AD \to E$

- For $A \to B$

  $(A^+)_{\text{when } A \to B \text{ is present}}$ = ABCE

  $(A^+)_{\text{when } A \to B \text{ is not present}}$ = AC

Since, we are getting different result so $A \to B$ is not redundant.

- For $B \to C$

  $(B^+)_{\text{when } B \to C \text{ is present}}$ = B C E

  $(B^+)_{\text{when } B \to C \text{ is not present}}$ = B E

Since, we are getting different result so $B \to C$ is not redundant.

- For $A \to C$

  $(A^+)_{\text{when } A \to C \text{ is present}} = A\ C\ B\ E$

  $(A^+)_{\text{when } A \to C \text{ is not present}} = A\ B\ C\ E$

  Since we are getting same result, so $A \to C$ is redundant, so eliminated.

- For $D \to E$

  $(D^+)_{\text{when } D \to E \text{ is present}} = D\ E$

  $(D^+)_{\text{when } D \to E \text{ is not present}} = D$

  Since we are getting different result, So $D \to E$ is not redundant.

- For $B \to E$

  $(B^+)_{\text{when } B \to E \text{ is present}} = B\ E\ C$

  (B^+) when $B \to E$ is not present $= B\ C$

  Since, we are getting different result. So $B \to E$ is not redundant.

- For $AD \to E$

  $(AD^+) = ADEBC$

  $(A^+) = ABCE$

  $(D^+) = DE$

  From above closure, we can conclude that $AD \to E$ is not redundant.

  $\therefore$ Required canonical cover

  $A \to B,\ B \to C,\ D \to E,\ B \to E,\ AD \to E$

Now,

$r\ (A\ \ B\ \ C\ \ D\ \ E)$

Clearly, from the above diagram we conclude that (A, B) is a candidate key.

Prime attributes $\Rightarrow A, D$

Non-Prime attributes $\Rightarrow B, C, E$

The functional dependencies $A \to B$, $D \to E$ are partial dependencies and $B \to C$, $B \to C$ are transitive dependencies

Thus, it is not in 2NF also.

Noomalising

removing partial dependencies

$r\ (\underline{A}\ \ B\ \ C\ \ D\ \ E)$

$r\ (D, E)$

$r\ (A, D)$

Removing transitive dependencies

r (A B)

r (B C E)

r (D E)

r (A D)

The above decomposition is clearly in 3NF because → The relational schema R is third normal if, whenever a non trivial functional $X \rightarrow A$ holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R.

∴ Required Normalisation ⇒ r(AB), r(BCE), r(DE), r(AD)

Khushi

DATABASE MANAGEMENT SYSTEMS

ETCS - 208

ASSIGNMENT - 4

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

MAIT

उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

Faculty : Dr. Anu Rathee

Submitted by : Khushi
Roll No : 03114813120
Semester : 4th
Branch : ITE
Group : ITE123

Khushi
03114813120

Ques Find the id and title of all courses which do not require any prerequisites.

Ans Select course_id, title from course where course_id not in (select course_id from prereq);

Ques Find the names of students who have not taken any biology dept courses

Ans. Select name from student
where ID in (Select ID from
where course_id in (select course_id from cousse
where dept_name != 'Biology'));

Ques Write SQL update queries to perform the following

(i) Give a 10% hike to all instructors

update instructor set salary = salary * 1.1;

(ii) Increase the tot_credits of all students who have taken the course titled "Genetics" by the number of credits associated with that course.

update students set tot_credit = tot_credit + (select credits from course
where course_title = "Genetics").

where ID in (select ID from student, course
where student.dept_name = course.dept_name and course title = 'Genetics');

(iii) For all instructors who are advisors of at least 2 students, increase their salary by 50000.

update instructor set salary = salary + 50000
where name in (select name from instructor, advisor
where instructor.ID = advisor.ID
group by instructor.name having count (s-ID) > 1);