NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur

Affiliated to Pokhara University

ASSIGNMENT FOR DATABASE MANAGEMENT SYSTEM

ASSIGNMENT 4

Submitted by:
Name: Arpan Adhikari
Roll No.: 231309
BE-CE (3rd SEM)

Submitted to:
Er. Simanta Kasaju

Name : Arpan Adhikari
Roll No. : 231309
BECE (Day)

Assignment 4

Q1) compare trivial and non-trivial dependency.

→ Both trivial and non-trivial dependency are concepts used to describe functional dependencies within a database.

	Aspect	Trivial dependency	Non-trivial dependency
1.	Definition	A functional dependency $X \rightarrow Y$ is trivial if $Y \subseteq X$.	A functional dependency $X \rightarrow Y$ is non-trivial if $Y \not\subseteq X$.
2.	Example	For A and B, $AB \rightarrow A$ (because $A \subseteq AB$)	For A and B, $A \rightarrow B$ (because $B \subseteq A$)
3	Implication	Trivial dependencies are automatically true and donot provide new information about the data	Non-trivial dependencies represent meaningful relationships and constraints betⁿ attributes.
4.	Normalization Impact	Trivial dependencies are not a concern for normalization because they don't vidate any normal form.	Non-trivial dependencies are critical for normalization they help identify and eliminate redundancy in database design.
5.	Use case	Often ignored in practical database design as they are obvious and redundant.	Used to determine key attributes and eliminate anomalies during normalization.
6.	Complexity	Simple to identify and does not affect database structure.	Requires careful analysis to ensure the database is properly normalized.

Q2) Discuss the un-normalized form with examples.

Un-Normalized Form (UNF):-
A database table is in UNF if it contains redundant data, multi-valued attributes, or lacks structure based on normalization rules.

Example: StudentInfo

StudentID	Name	Courses	PhoneNO.
101	Alice	Math, Science	9864389333, 9742886222
102	Bob	English	9846376021

Problems:
1. Multi-valued attributes
Courses and PhoneNO. have multiple values in one cell.

2. Redundancy
Inefficient storage due to repeated data.

3. Anomalies
Difficult updates and insertions. For example, adding a new course for Alice is complen

Q3)

→

Given, R(X, Y, Z, W)

Set of F.D. are:

$X \rightarrow Y$

$Y \rightarrow Z$

$Z \rightarrow X$

$(X)^+ \rightarrow XYZW$

$(WY)^+ \rightarrow YZX \cdot W$

$(WZ)^+ \rightarrow ZXYW$

C.K. = WX, WY, WZ

Q4)

→ Given, R(X, Y, Z, W)

set of F.D. are:

PQ → R XY → ZW

R → S W → X

Q → PT

Here, $Y^+ \longrightarrow Y$

Now, $(XY)^+ \longrightarrow XYZW$ ✓

$(YZ)^+ \longrightarrow YZ$

$(YW)^+ \longrightarrow YWXZ$ ✓

candidate key = XY, YW Ans

Q5)

Soln

$PQ \rightarrow R$

$R \rightarrow S$

$Q \rightarrow PT$

Now, $Q^+ \rightarrow QPTRS$

$(U)^+ \rightarrow U$

Now,

$(QU)^+ \rightarrow QUPTRS$

$\therefore CK = QU$

No. of CK = 1.

Q6)

→

The attribute closure of a set of attributes A is the set of attributes that can be functionally determined by A using a given set

of functional dependencies.

Steps to calculate Attribute closure:

1. start with $A^+ = A$ (the attributes in A)
2. For each functional dependency $X \rightarrow Y$:
   If $X \subseteq A^+$, add Y to A^+
3. Repeat until no new attributes can be added.

→ A FD is a constraint between two sets of attributes in a relational database.
It is represented as $X \rightarrow Y$

	Aspect	F.D.	Partial D
1.	Definition	A functional dependency $X \rightarrow Y$ is full if all attributes of X are required to determine Y.	A functional dependency $X \rightarrow Y$ is partial if some attributes of X can determine Y
2.	Example	$AB \rightarrow C$ Both A & B are needed to determine C	$AB \rightarrow C$ if $A \rightarrow C$ then, $AB \rightarrow C$ is partial
3.	Dependency on C.K	Occurs when Y is dependent on the entire candidate key	Occurs when Y is dependent on only a subset of C.K.
4.	Normalization impact	Appears in relations in 2NF or higher.	Must be removed to move from 1NF to 2NF.

Q7)

→ Decomposition of a relational schema involves breaking a single relation R into two or more smaller relations $R_1, R_2, \ldots, R_n$ in such a way that the smaller relations:

- Preserve the data and semantics of the original relation.
- Eliminate redundancy and anomalies.
- Are in higher normal forms (e.g. 2NF, 3NF, BCNF)

Soln:

R(ABCGHI)

Set of FD are: $A \rightarrow B$
$A \rightarrow C$
$CG \rightarrow HI$
$B \rightarrow H$

Here, $(A)^+ \rightarrow ACB$
~~$(B)^+ \rightarrow BH$~~
$(G)^+ \rightarrow G$

Now, ~~$(AB)^+$~~
$(ABG)^+ \rightarrow ABGCHI$ ✓.

Q8)

→

a) $\{W \rightarrow Y, X \rightarrow Z\} = \{WX \rightarrow Y\}$

Soln

Here, $W \rightarrow Y$

Y is already determined by W. Adding X (an independent attribute) doesn't impact Y.
So, $WX \rightarrow Y$ is valid due to augmentation.

b)

$X \rightarrow Y$

$XW \rightarrow W$

$WY \rightarrow Z$ To prove, $X \rightarrow Z$

Soln

Combining $X \rightarrow Y$ and $X \rightarrow W$, (IR4)

$X \rightarrow WY$ — (i)

Given, $WY \rightarrow Z$ — (ii)

From (i) & (ii), using Transitive Rule,

$X \rightarrow Z$ Proved.

c) $X \rightarrow Y$ -(i)

$XY \rightarrow Z$ -(ii) To prove, $X \rightarrow Z$

Soln

Used Pseudo transitive rule in (ii) using (i),

$XX \rightarrow Z$

or, $X \rightarrow Z$

Proved.

Q9)

Soln

R(PQRSTUV)

FD are: $P \rightarrow Q$

$QR \rightarrow ST$

$PTV \rightarrow V$

$(QR)^+ \rightarrow QRST$

$(PR)^+ \rightarrow PRQST$ Ans

Q11)

→ Function Dependency (FD) equivalency occurs when two sets of functional dependencies imply the same constraints on database schema. That means, if two sets of FDs F and G are equivalent, any FD derivable from F can also be derived from G and vice versa.

Example:

Let, F = A → B, B → C
G = A → C, A → B

Verifying:

1. From F
A → B is given.
A → C can be derived (since A → B and B → C)

2. From G
A → B and A → C are given.
B → C can be derived.

Both sets lead to same results, so they are equivalent.

Q12)

→ Given, R(X, Y, Z, W, V)

FD for P:
X → Y
XY → Z
W → XZ
W → V

FD for Q:
X → YZ
W → XV

Q15)

→

Redundancy isn't always bad - it can improve performance (e.g. denormalized databases). However, uncontrolled redundancy causes issues like:

- Update anomalies: Data inconsistency during changes.
- Wasted storage: Repeated data uses space unnecessarily.
- Complex maintenance: Harder to manage data.

Example:

Non-Normalized Table:

StudentID	Course	Instructor	Email
101	Math	Smith	smith@ncit.edu.np
102	Math	Smith	smith@ncit.edu.np

Problem: Changing Smith's email requires updating all rows.

Solution (Normalized):

1. Students Table

StudentID	Course	Instructor
101	Math	Smith

2. Instructors Table

Instructor	Email
Smith	smith@ncit.edu.np

Benefit of Redundancy:
- Data safety in case of hardware failure.
- Backup server allows availability always.

Q16)

→

Third Normal Form (3NF) is a database normalization rule that ensures a database is free of certain types of redundancy and dependencies.

For being in 3NF,
→ It must be in 2NF
→ It has no transitive dependencies.

ED from table :

Address ⟶ Phone ✗3NF
salary ⟶ Post ✗3NF
Name ⟶ Phone, Address, salary, Post ✓3NF.

(Name)$^+$ ⟶ Name, Address, Phone, salary, Post

CK = Name
PA = Name
NPA = Address, Phone, salary, Post

R_1 (Address Phone) R_2(salary Post) R_3(Name)

R_1 :

Address	Phone
KTM	11111
BKT	22222

R_2 :

salary	Post
20000	Engineer
10000	Officer
15000	Overseer

R_3 :

Name
A
B
C
D
E

Q17)

1NF

Raghukul
Date
Page No.

Simplest normal form.

Rules:
- Column must be atomic i.e. no multi-valued attributes
- Table must have single valued attributes

Eg:-

A	B	C
a1	b1	c1
a2	b2	c2

⇓

A	B
a1	b1
a2	b2

A	C
a1	c1
a2	c2

2NF

Rules:
- Must be in 1NF.
- No

3NF

BCNF

4NF

5NF

Q18)

→

Aspect	3NF	BCNF
1. Definition	A table is in 3NF if it is in 2NF and all non prime attributes are non-transitively dependent on P.K.	A table is in BCNF if it is in 3NF and for every functional dependency, the determinant is C.K.

Aspect	3NF	BCNF
2. Key concept	3NF removes transitive dependencies (non-prime attributes depending on other non-prime attributes)	BCNF removes anomalous functional depencies when a non-candidate key attribute functionally determines another attribute.
3. Functional Dependency	It allows cases where a non-prime attribute determines another non prime attribute.	It dissolves such cases: every determinant must be CK.
4. ~~When~~ Problems arising cond^n	If a NPA depends on another NPA.	If any attributes determine NPA, and its not CK.

Q19)

a) → Yes, there is an insertion anomaly. If a new department (e.g. D005) needs to be added, it cannot be done without assigning it to at least one employee. This is because the emp-id, emp-name and emp-address are part of the table structure, and they must have a value to insert a new row. (No null for emp-id (PK))

b) → Yes, there is a modification anomaly. If an employee's address changes (e.g. Rick's address changes from Delhi to Mumbai), it must be updated in all rows where emp-id = 101. If one row is missed during the update, it will lead to inconsistent data.

c) → Yes there is a deletion anomaly. If all employees belonging to a department (e.g. D890 for maggie) are deleted, the information about that department (D890) will also be lost, as there is no separate table to store department information independently.

Q20)

a) FDs are:

Shipment ⟶ Origin, Destination, Distance

Origin → Distance
Destination → Distance

b) → ~~The closure of shipm~~

Here,

$(shipment)^+$ → shipment, origin, Destination, Distance

CK = shipment
PA = shipment
NPA = Origin, Destination, Distance

For shipment,

	BCNF	3NF	2NF	1NF
shipment → Origin, Destination, Distance	✓	✓	✓	✓

"Shipment" is in Boyce code Normal form along with 3NF, 2NF & 1NF.

c) → Yes, there is insertion anomaly.
For e.g. If a new route with a known distance needs to be added, we cannot do so unless there is a shipment assigned to that route. This dependency on shipment for storing route data is unnecessary.

d) → Yes, there is modification anomaly.
For e.g. If the distance between seattle and Denver changes, it needs to be updated in multiple rows. Failure to update all rows can result in inconsistencies.

e) → Yes, there is deletion anomaly.
For e.g., if the shipment 409 is deleted, the information about the distance between seattle and Denver will also be lost. This happens because route data is tied to shipment data.

f) → "shipment is already in 3NF".

Q21)

→ Reason for Normalization:

(i) Eliminate Redundancy
(ii) Avoid Anomalies
(iii) Ensure data integrity
(iv) Enhance query performance

Q22) Table conversion:

→

a) 1NF (no multivalued attributes)

Full Names	Physical Address	Movies Rented	salutation
Janet Jones	First street Plot No 4	Pirates of Caribbean	Ms.
Janet Jones	First street Plot No 4	Clash of Titans	Ms.
Robert Phil	3rd street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	clash of Titans	Mr.

b) 2NF

FD are: ~~Janet Jones~~ FullNames → salutation

PhysicalAddress → FullNames

23

Aspect	Lossless Dependency	Lossy Dependency
1. Definition	No data is lost after decomposition and recombination	some data is lost after decomposition and recombination
2. Integrity	Maintains data integrity.	Loses data integrity
3. condition	At least one common attribute must act as a superkey.	common attributes do not form a superkey.
4. Example	Correctly normalized tables (e.g. 3NF)	Poorly designed decomposition.

Q24)

Sol^n books (accessionno, isbn, title, author, publisher)
users (userid, name, deptid, deptname)

		3NF
books	accessionno → isbn	✓
	isbn → title	x
	isbn → publisher	x
	isbn → author	x
users	userid → name	✓
	userid → deptid	✓
	deptid → deptname	x

For books table, ~~Ass~~ Assume it is in 2NF,

(accessionno)$^+$ → accessionno, isbn, title, publisher, author

CK = accessionno

PA = accessionno

NPA = isbn, title, publisher, author

Decomposition:

R_1 (isbn title), R_2 (isbn publisher), R_3 (isbn author), R_4 (accessionno)

For user table,

$(userid)^+ \longrightarrow$ userid, name, deptid, deptname

CK = userid

PA = userid

NPA = name, deptid, deptname

Decomposition:

R_1 (deptid deptname) R_2 (userid)

(Q27)

→ R(A, B, C, D)

FD sets are: 2NF

$AB \rightarrow CD$

$B \rightarrow C$

Solution: Decompose into

$R_1(BC)$

$R_2(ABD)$

Now, R_1 and R_2 are in 2NF.

36)

→ Higher normal forms like 4NF and 5NF address are rare anomalies that occur in very specific cases.

→ Ensuring BCNF typically eliminates most redundancy and anomalies without excessive decomposition.

→ Practical designs prioritize simplicity, performance and ease of querying over strict adherence to higher normal forms.

37)

→

The three design goals are:

1. Minimizing Redundancy

2. Avoiding Anomalies

3. Ensures data integrity.

38)

→ De-normalization is the process of restructuring a normalized database to improve query performance by intentionally introducing redundancy.

This is done by combining tables or adding redundant data to reduce the no. of joins required during queries.

While normalization minimizes redundancy and ensures data integrity, it can cause performance issues for read-heavy operation due to:

- Increased number of joins in queries
- Higher latency due to frequent lookups.

Achieving De-normalization as:

- Adding Redundant Data
- Combining Tables
- Storing pre-aggregate Data
- Creating Summary tables
- Using indexed views

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur

Affiliated to Pokhara University

ASSIGNMENT FOR DATABASE MANAGEMENT SYSTEM

ASSIGNMENT 7

Submitted by:
Name: Arpan Adhikari
Roll No.: 231309
BE-CE (3rd SEM)

Submitted to:
Er. Simanta Kasaju

Name : Arpan Adhikari
Roll No. : 231309
BECE (Day)

Assignment 7

Q1) employee(empname, street, city)
works(empname, companyname, salary)
company(companyname, city)
manages(empname, managername)

(i) create table employee(
empname varchar(20),
street varchar(20),
city varchar(20));

(ii) Insert into works
values
('Arpan', 'Microsoft', '10000000');

(iii) select empname, city from employee join works
on employee.empname = works.empname
where companyname not like 'XYZ Pvt. Ltd.';

Q2) 2022 Fall

(i) create table Actress_details (
Players_id int primary key,
Actress_name varchar(20),
Debut_year int,
Recent_release varchar(20),
Actress_fee ~~long~~ bigint);

(ii) delete from Actress_details
where Recent-Release = 'Prem';

(iii) update table Actress_Details
set Recent_release = 'Win the Race'
where Actress_name = 'Renu';

(iv) insert into Actress_Details
values
(1, 'Rani', 2010, 'sundari', 20000);

Q3) 2021 spring

(i) select * from sailors join Reserves
on Sailors.sid = Reserves.sid
where bid = 103;

(ii) Update table Boats
set color = 'green' where bid = 104;

(ii) Select sname from sailors jon Reserves
on sailors.sid = Reserves.sid where bid = 103 and day = 5;

(iii) Select sname from sailors join Reserves
on sailors.sid = Reserves.sid
join Boats on boats.bid = Reserves.bid
where color = 'red' or color = 'green'.

(iv) Select * from sailors where name != 'Ram';

(vi) select bname from Boats;

Q4) 2021 fall

i) create table chef (
chef-license int primary key,
c-fname varchar(30),
c-lname varchar(20),

c_dob date,
c_gender varchar(20),
c_experience_hours int,
c_photograph BLOB);

ii) Insert into chef
values
(1010, 'Arpan', 'Adhikari', [redacted] '2006-3-9', 'Male', '30', 'Img2.jpg');

iii)
update table chef
set c_experience_hours = 3
where c_fname = 'Naresh';

iv) Delete from chef
where c_fname like '_%or';

v) select c_experience_hours from chef;

vi)

vii) select * from chef
order by C_lname DESC, C_fname ASC;

Q7)

i) create database hotel;
use hotel;

create table Hotel_details(
hotel_id int primary key,
hotel_name varchar(30),
estb_year int,
hotel_star int,
hotel_worth int);

ii)

iii) Update Hotel_details
set hotel_star = '4 where hotel_name = 'Hotel Chitwan';

iv) Delete from Hotel_details
where hotel_worth > 9;

Q10)

1) Select Name from Doctor
where Name not in (select name from work);

2) Update Doctor
set Address = 'Pokhara'
where Name = 'Dr.Hari';

3) Delete from works
where Depart_no in (select Depart_no from Department where
Depname = 'OPD');

4) Select Name
from works
group by Name
having count (Distinct Depart_no >=2);

Q14)

i) create table Teacher (
Teacher-ID int primary key,
TeacherName varchar(30),
Office varchar(30));

ii) Selet * from teacher;

iii) Select distinct * from teacher;

iv) Alter table teacher
Add Gender varchar(20);
column

v) Select * from teacher
orderby teachername asc;

vi) Delete from teacher ... ?

Q16)

i) select count (Distinct driver_Id)
from participated
where report_number in (
select report_number
from accident
where year(date) = 2015);

ii) select count(*) from participated
where driver_id in (select driver_id from person
where name = 'Peter Parker');

(iii) Delete from car
where model = 'Range Rover'
and license in (
select license from owns
where driver-id in (
select driver-id
from person
where name = 'John smith'
)
);

(iv) Update participated
set damage_amount = 3000
where car = 'ACD9021J3' and report_number = 'R27';