ASSIGNMENT 1

1) (A) Let $X = \Pi_{C}(\gamma_{\text{COUNT(genre)}}(\text{Movie}))$ Denotes the Count of Genre

FINAL QUERY

$$\Pi_{\text{Customer\_id}, \text{name}}(\sigma_{C=X}(\Pi_{\text{customer\_id}, C, \text{name}}({}_{\text{customer\_id}}\gamma_{\text{Count (genre)}} \text{ as } C(\sigma_{\text{city}=\text{Boston}}(\text{Customer} \bowtie \text{Rental} \bowtie \text{Movie})))))$$

B)

$$\Pi_{R, \text{movie-id}}((\underbrace{\Pi_{R, \text{movie\_id}}({}_{\text{movie\_id}}\gamma_{\text{avg(Rental\_price)}} \text{ as } R(\text{Movie} \bowtie \text{Rental}))}_{\text{movie\_id with avg rental rate}})) \bowtie \underbrace{(\Pi_{\text{movie\_id}}(\sigma_{C>10}({}_{\text{movie\_id}}\gamma_{\text{count(Rental id)}} \text{ as } C(\sigma_{\text{releaseyear}<2000}(\text{Movie} \bowtie \text{Rental})))))}_{\text{movie id with Rentalcount > 10 and Released before 2000}}$$

2)

A)

$$\{ E.\text{FName} \mid (E \in \text{Employee}) \land \exists W, P \, ((P \in \text{Project}) \land (W \in \text{WORKS\_ON}) \land (P.\text{project id} = W.\text{Project id}) \land (W.\text{Emp\_id} = E.\text{Emp\_id}) \land (P.\text{dept\_id} = 5)) \}$$

B) Considering the Emp_id in Department to be the Manager.

$$\{ E.\text{Fname} \mid (E \in \text{Employee}) \land \exists D, DE \, ((D \in \text{Department}) \land (DE \in \text{Dependent}) \land (D.E.\text{Eid} = D.\text{Eid})) \}$$

Assignment 2: Views,Transactions, Roles and Authorizations

(1) Consider the following relation : Movie (id : integer, title : string, genre : string, release_year : integer). Suppose you have a view called "movies_2021" defined as follows :
CREATE VIEW movies_2021 AS
SELECT * FROM Movie WHERE release_year = 2021 ;
Explain what happens when you try to execute the following insertion query on the "movies_2021" view :
INSERT INTO movies_2021 (id , title , genre , release_year) values (203, 'RRR', 'Drama' , 2022) ;

When i tried inserting in postgres it worked fine,that means this insertion was reflected in both the view and the table , but the view is defined with a filter condition that restricts the rows to those with a release year of 2021, and the insertion query is attempting to add a new row with a release year of 2022 and that violates the filter condition .

(2) Consider the following customers relation , initially with no records : customers (customer_id , customer_name, city)
Suppose the following SQL queries are executed in order :
1) INSERT INTO customers (customer_id, customer_name, city) VALUES (1, 'John Smith', 'Boston') ;
2) SAVEPOINT s ;
3) UPDATE customers SET city = 'Los Angeles' WHERE customer_id = 1 ;
4)INSERT INTO customers (customer_id, customer_name, city) VALUES (2, 'Jane Doe', 'New York') ;
5) ROLLBACK TO s ;
6)UPDATE customers SET city = 'San Francisco' WHERE customer_id = 1 ;
7)COMMIT ;
If all the above SQL queries run in order, what will be the result of the below query ?
SELECT * FROM customers ;
Give an explanation for your answer.

ASSUMING BEGIN WAS DONE BEFORE LINE 1_
Then since there is a rollback done before commit on line 7, only effects of line1,6 will be seen on the table.
And the result of the query will be

```
  1    john smith    San Francisco
```

IF BEGIN WAS NOT DONE BEFORE LINE 1
If begin is not done then every statement in postgres is considered a begin - commit block so the rollback on 5 will not make sense , hence the result of the query will be

```
1    john smith     San Francisco
2    jane doe       New York
```

(3) Suppose a company has a database that contains information about its sales and customers. The sales manager in the UK needs access to the sales information for all customers based in the UK. However, the sales manager in the UK should not have access to the sales' information for the customers based in other countries. As a database administrator, how can you ensure data security by providing the appropriate level of access to the relevant information ?

For this situation we can create a view for the information in Uk and Grant select on this view to the manager.now he won't be able to access the information of other countries.

**(4) Suppose there is a database with the following relation :
employees(id : integer , name : string, salary : real , age : integer)
The following command is executed :
GRANT UPDATE (id, name, age) ON employees TO John ;
if John executes the following command :
UPDATE employees SET name = ’Alice’, id = 50 WHERE id = 101 ;
Will the update be successful or not ? Justify your answer.**

Assignment 3: Functions, Procedures, Triggers

(1) Consider the ’Department’ relation containing the attributes ’dept_name’ and ’budget’. Show the details of the departments which have budgets more than the average budget across all departments by defining a function ’more_than_avg_budget’ in SQL.

```
CREATE OR REPLACE FUNCTION more_than_avg()
RETURNS TABLE(dept_name varchar) AS
$$
BEGIN
  RETURN QUERY SELECT dept_name FROM Department
             WHERE budget > (SELECT AVG(budget) FROM Department);
END;
$$ LANGUAGE plpgsql;
```

(2) Consider the ’Student’ relation containing the attributes ’ID’, ’Dept_name’, ’Credits’. Create a procedure that deletes all students who are having ’Credits’ less than 5.0 in the ’CS’ department using SQL statements.

```
CREATE OR REPLACE PROCEDURE delete_low_credit_cs_students()
LANGUAGE plpgsql
AS $$
BEGIN
   DELETE FROM Student
   WHERE Dept_name = 'CS' AND Credits < 5.0;
END;
$$;
```

(3) How do DBMS automatically handle the condition mentioned in the previous question and meet the data consistency.

Either cascade or triggers can be used to maintain data consistency .

ASSIGNMENT 4

1)

$\rightarrow \{A \rightarrow B,\ C \rightarrow B,\ D \rightarrow ABC,\ AC \rightarrow D\}$

$\rightarrow \{A \rightarrow B,\ C \rightarrow B,\ D \rightarrow A,\ D \rightarrow B,\ D \rightarrow C,\ AC \rightarrow D\}$

Now for each dependency we check if it is redundant or not.

(1) $A \rightarrow B$

with dependency	after removing $A \rightarrow B$
$A^+ = \{A, B\}$.	$A^+ = \{A\}$

So $A \rightarrow B$ is not redundant

(2) ~~A~~ $C \rightarrow B$

with $C \rightarrow B$	without $C \rightarrow B$
$C^+ = \{C, B\}$	$C^+ = \{C\}$

So $C \rightarrow B$ is not redundant

(3) $D \rightarrow A$

with $D \rightarrow A$	without $D \rightarrow A$
$D^+ = \{A, B, C, D\}$	$D^+ = \{B, C, D\}$

Hence $D \rightarrow A$ is not redundant.

D -> B

With D -> B	without D -> B
$D^+ = \{A, B, C, D\}$	$D^+ = \{A, B, C, D\}$

hence D -> B is redundant and can be removed.

D -> C

with D -> C	without D -> C
$D^+ = \{A, B, C, D\}$	$D^+ = \{A, B, D\}$

hence D -> C is not redundant.

AC -> D

with AC -> D	without AC -> D
$AC^+ = \{A, B, C, D\}$	$AC^+ = \{A, B, C\}$

hence AC -> D is not redundant

also A^+ doesnot contain C, and C^+ does n't contain A so AC cant be brokendown.

So minimalized version is

{ A -> B, C -> B, D -> A, D -> B, AC -> D }

(2) $F_1 = \{A \to B, AB \to C, D \to AC, D \to E\}$

$F_2 = \{A \to BC, D \to AB, C \to B\}$

to check equivalence let's compare A^+, B^+, C^+, D^+ for Both

F_1

~~A~~ = (1) A^+

1) $A \to A$ {trivial}

2) $A \to B$ {Given}

3) $AB \to C$ {Given}

4) $A \to AB$ {by (1), (2)}

5) $A \to C$ {by transitivity on 3, 4}

$A^+ = \{A, B, C\}$

(2) C^+

) $C \to C$ {trivial}

$C^+ = \{C\}$

$C^+ = \{C\}$

F_2

A^+

$A \to A$ {trivial}

$A \to BC$ {Given}

$A^+ = \{A, B, C\}$

C^+

1) $C \to C$ (trivial)

2) $C \to B$ (Given)

$C^+ = \{C, B\}$

So F_1, F_2 are not equivalent because C^+ is F_1 and F_2 are not same

Q(3) FD : $\{BC \to D, C \to A, D \to B\}$

Candidate keys are BC, DC

Prime attributes are B, C, D

because of $C \to A$, it is a partial dependence So not even 2NF

So a decomposition of R_1 (B,C,D), R_2 (C,A), R_3 (D,B) will make a 3NF decomposition

R_1 (BCD), FD = $\{BC \to D\}$ it is 3NF

R_2 {C, A}, FD = $\{C \to A\}$ it is 3NF

R_3 (D, B), FD = $\{D \to B\}$ it is 3NF

Q.4 $R(X,Y,Z)$ FD := $(XY \to Z, Z \to Y\}$

→ Candiatake Keys are XY, XZ

→ it doesnot Satisfy BCNF because of $Z \to Y$
Z is not a candidate key

→ 3NF is the Highest it is Satisfying.

→ We decompose R into $R_1(X,Z)$, $R_2(Z,Y)$
and that is in BCNF

→ this decomposition is lossless because Z is candidate key in R_2. and it is the intersection of R_1 ; R_2

→ its is not dependency preserving because the dependency ~~thisiea~~ $XY \to Z$ is not preserved.