ASSIGNMENT 3

QUESTION 1

Start.php

```
<!DOCTYPE html>
<html>
<body>
<form action = "search.php" method = "post">
Enter a String: <input type = "text" name = "name">
<br>
<br>
Submit:<input type = "submit" name = "search">
</form>
</body>
</html>
```

Search.php

```
<!DOCTYPE html>
<html>
<head>
</head>

<body>
<form method="POST" action="attri.php">
<select name="name">
<?php

$conn=mysqli_connect('localhost','root','','information_schema');
$sql = "SELECT DISTINCT(TABLE_NAME) FROM INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME LIKE '%".$_POST['name']."%' and table_schema='studuniv'";
$result = mysqli_query($conn, $sql);
while($row = mysqli_fetch_assoc($result))
{
echo '<option value ='.$row['TABLE_NAME'].'>'.$row['TABLE_NAME'].'</option>';
}
echo '</select>';
?>
<br><br><input type="submit" value="Submit">
</form>
</body>
</html>
```

Attri.php

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<form method="POST" action="table.php">
<select name="attribute_name">
<?php

$conn=mysqli_connect('localhost','root','','information_schema');
$sql = "SELECT column_name from information_schema.columns where
table_schema='studuniv' and table_name ='$_POST[name]'";
$result = mysqli_query($conn, $sql);
while($row = mysqli_fetch_assoc($result))
{
echo '<option value ='.$row['column_name'].'>'.$row['column_name'].'</option>';
}
echo '</select>';
?>
<br><br><input type="submit" value="Submit">
</form>
</body>
</html>
```

Table.php

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<form method="POST" action="attri.php">
<select name="name">
<?php
$conn=mysqli_connect('localhost','root','','information_schema');
$sql = "SELECT TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE
COLUMN_NAME
='$_POST[attribute_name]' and table_schema='studuniv'";
$result = mysqli_query($conn, $sql);
while($row = mysqli_fetch_assoc($result))
{
echo '<option value ='.$row['TABLE_NAME'].'>'.$row['TABLE_NAME'].'</option>';
```

```
}
echo '</select>';
?>
<br><br><input type="submit" value="Submit">
</form>
</body>
</html>
```

QUESTION 2

1)

```
SELECT lname, year(payment_date),SUM(penalty) FROM residence natural join Landlord
natural join LeasePayment GROUP BY year(payment_date);
```

2)

```
create trigger penalty after insert on LeasePayment
referencing new row as nrow
referencing old row as orow
for each row
when (nrow.day(payment_date) > 5 )
begin
 set nrow.penalty= (orow.penalty + 5* (nrow.day(payment_date)-5))
end;
```

3)

```
Create trigger status_check before update on Leases
referencing new row as nrow
For each row
If (nrow.startdate< NOW() and nrow.endtable>NOW())
        Then set Residence.status='Rented' where Residence.rid=nrow.rid
Else
        Set Residence.status='Available'
End;
```

QUESTION 3

a) This is not a good design for following reasons:

1) It doesn't follow the normalization rules. As everything is in one table it makes maintenance very difficult
2) Al the data is stored in one huge table which makes querying very time consuming.

3) Inserting data into the table would require columns to have NULL values. The data would require more storage and be inconsistent.
4) The data shows a lot of functional dependencies. Therefore there is a lot of information being repeated in the tuples.

(b)

The non-trivial functional dependencies F for this schema are :-

{pid} -> {pname, page, pstate, admittime}
{docid} -> {docname, docage,doclevel,docsalary}
{did} -> {dname, dtype, description}
{rid}-> {rname, rcapacity}
{doclevel} -> {docsalary}
{rid}->{dtype}
{pid,did,docid,daysadmitted}->{tcost}
{patienttype,tcost} ->{discount}

(c)

The canonical cover for the functional dependencies in F is:

{pid} -> {pname, page, pstate,admittime}
{docid} -> {docname, docage, doclevel}
{doclevel} -> {docsalary}
{did} -> {dname, dtype, description}
{rid}-> {rname, rcapacity,dtype}
{pid,did,docid,daysadmitted}->{tcost}
{patienttype,tcost} ->{discount}

(d)

The above schema is not in BCNF as a nonprime attribute can find a prime attribute, also there is a transitive dependency in the above relation. For schema to be BCNF for each non-trivial dependency A->B, A should be a superkey, which is not the case here.

The BCNF form is:

Patient (pid, pname, page, pstate, admittime)
Doctor (docid, docname, docage, doclevel)
DocSalary (doclevel, docsalary)
Disease (did, dname, dtype, description)
Room (rid, rname, rcapacity)
Cost (pid, did, docid, daysadmitted, tcost)

TotalDiscount (ptype,tcost,discount)
Treatment (pid, docid, did, rid, admittime, daysadmitted)