Freelancing DBMS



**Name :** Asfaria Islam Chowdhury

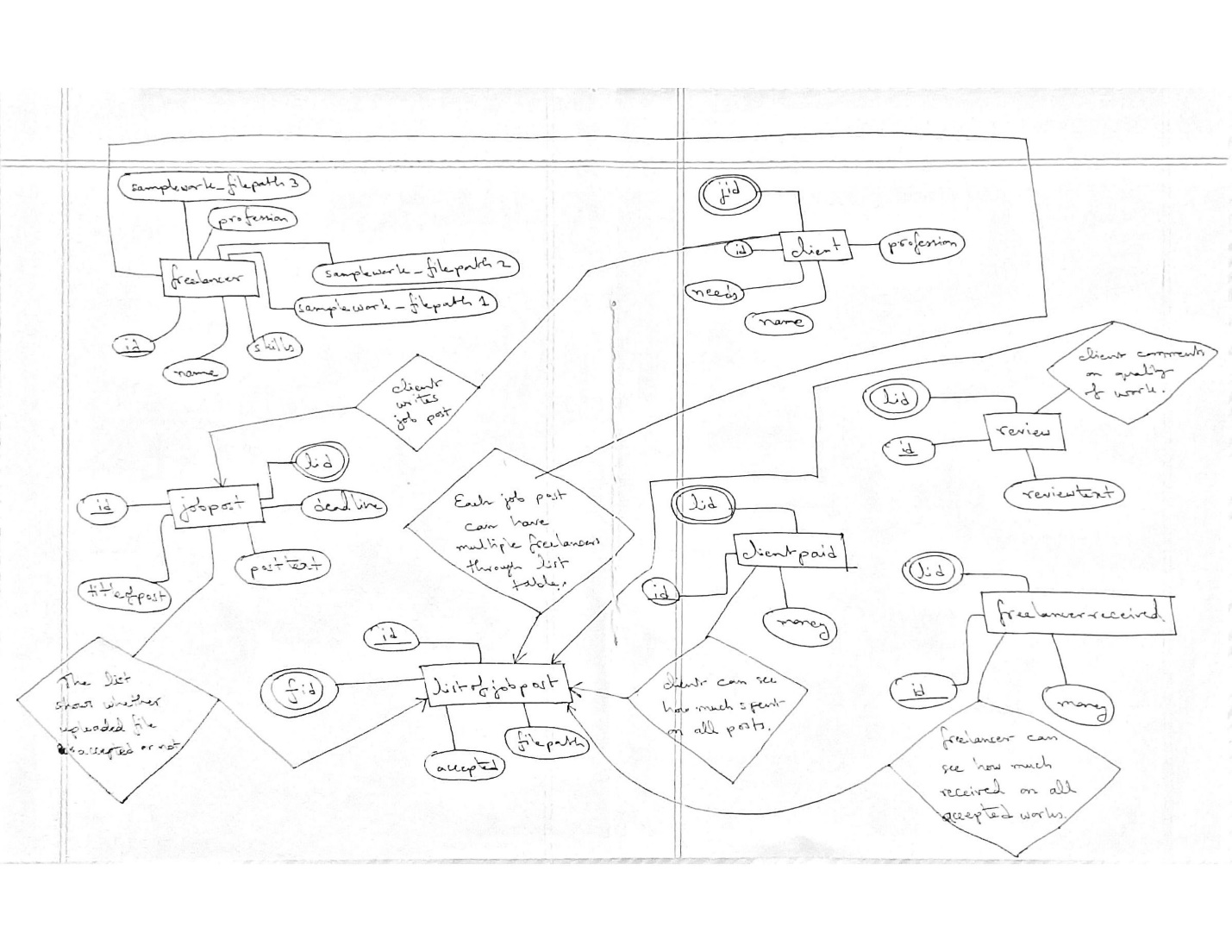
**ID :** 1931741642

**Course:** CSE311 Theory

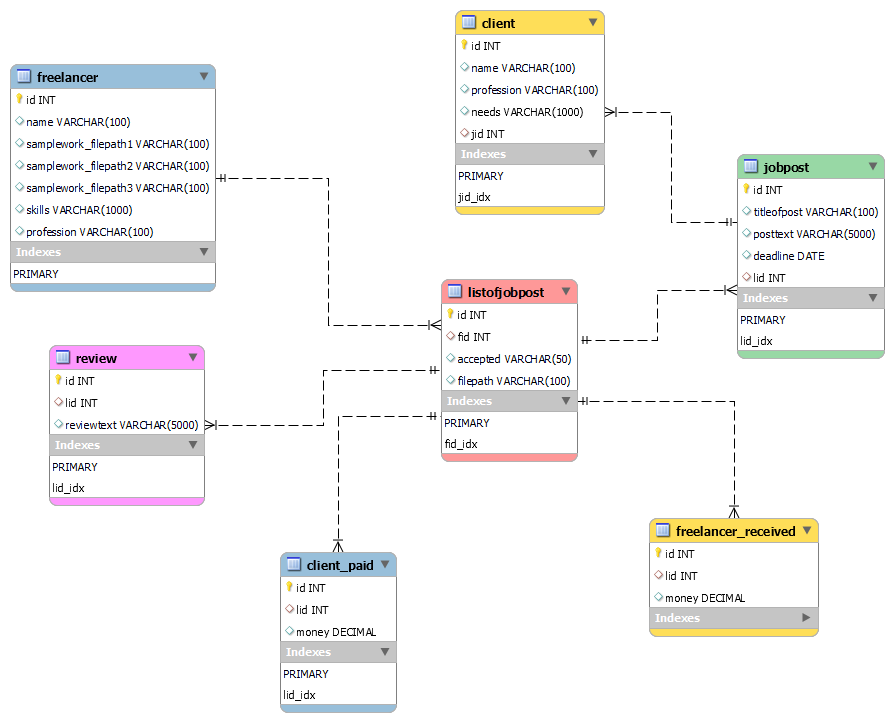
**Introduction**

In this project, a freelancing DBMS is built using Microsoft SQL. There are a total of 7 tables: freelancer, client, jobpost, listofjobpost, review, client\_paid, freelancer\_received.

**ER Diagram**

****

**DB Schema**

****

The freelancer table has three sample files to showcase clients their work, such as a content writer showing sample articles on food.

The client table has an attribute *needs* which stores the information on what the client is looking for, such as a Graphic Designer for making travel brochures. The foreign key *jid* refers to the id in jobpost that the client has posted. Only clients can post.

Each job post has a deadline of type DATE, and a foreign key *lid* which refers to listofjobpost table.

The listofjobpost is a table where multiple freelancers can apply to the same job post via the foreign key *fid* which refers to the freelancer table. The table has an attribute *accepted* which takes values ‘yes’ or ‘no’, depending on whether the client accepted or not. The attribute *filepath* simply refers to the file uploaded by the freelancer.

The review table has a foreign key *lid* to refer to the listofjobpost. In this way, multiple freelancers’ reviews can be viewed by multiple clients, via the *reviewtext* attribute.

The client\_paid table has *lid* as foreign key to refer to listofjobpost. The *money* paid by individual job posts can be viewed, where the value 0 refers to rejected work, i.e. *listofjobpost.accepted* = ‘no’ for a specific freelancer’s *listofjobpost.fid*. The table also helps viewing how much in total a client spent with the help of aggregate functions and joining onto tables.

The freelancer\_received is structured in a similar way. By using appropriate CASE statements, we can view freelancers into two categories of *payment\_status*, ‘high’ or ‘low’. This, in turn, shows which freelancer is more skilled, and helps clients look for the right freelancer.

**A Few Sample Code**

*Full code attached in separate file.*

**Code 1: We will join 4 tables to see *client.name*, *jobpost.posttext*, *listofjobpost.accepted*, *freelancer.name* – with condition that the freelancer is a designer, job post’s deadline is in the year 2021, OR the *accepted* status is ‘yes’. The table is order by client’s name (ascending).**

SELECT client.name, jobpost.posttext, listofjobpost.accepted, freelancer.name

FROM client JOIN jobpost ON client.jid = jobpost.id

JOIN listofjobpost ON jobpost.lid = listofjobpost.fid

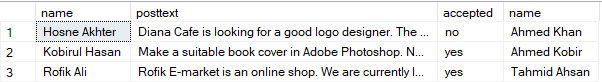
JOIN freelancer ON listofjobpost.fid = freelancer.id

WHERE freelancer.profession LIKE '%Designer%'

AND jobpost.deadline LIKE '2021-%'

OR listofjobpost.accepted = 'yes'

ORDER BY client.name;



**Code 2: We will be joining 3 tables and use CASE to view *payment\_status* of freelancer. For any specific work, *money* = 0 represents *listofjobpost.accepted* = ‘no’.**

SELECT freelancerreceived.money, listofjobpost.filepath, freelancer.name,

(CASE WHEN freelancerreceived.money >= 5000

then 'High'

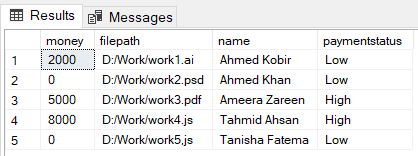
ELSE 'Low'

END

) AS paymentstatus

FROM freelancerreceived JOIN listofjobpost ON freelancerreceived.lid = listofjobpost.id

JOIN freelancer ON listofjobpost.fid = freelancer.id;



**Code 3: With the help of EXISTS and 2 wildcards (‘%Adobe%’ and ‘%Designer%’), we will see which designers use Adobe products.**

SELECT name FROM freelancer AS F1

WHERE skills LIKE '%Adobe%' AND EXISTS (

SELECT \* FROM freelancer AS F2

WHERE profession LIKE '%Designer%' AND F1.id = F2.id

);

