John Haney

COSC 641- Database Systems II

Spring 2021

Frostburg State University

# Blockchain Finance

# Database Final Project COSC 641 Part 1a

**Chapter 1**

1. **Employee\_data**

create view EMPLOYEE\_DATA as

select

distinct nvl(b.fname, '') || ' ' || nvl(b.mname, '') || ' ' || nvl(b.lname, '') as "name of employee",

b.street || ' ' || b.city || ', ' || b.city || ' ' || b.state as "address",

b.zip as "Zip code of Employee Address",

b.ssn as "SSN",

b.jobtitle as "Title",

to\_char(sysdate, 'YYYY') as "current year",

(select c.salary

from EMP\_ANNUAL\_DATA c

where c.year <= to\_char(sysdate, 'YYYY')

and c.empid = b.empid

and rownum = 1) as "current yearly salary",

(select c.taxdeduction

from EMP\_ANNUAL\_DATA c

where c.year <= to\_char(sysdate, 'YYYY')

and c.empid = b.empid

and rownum = 1) as "current year tax deduction",

(select g.branchstartdate

from branch\_employee g

where g.empid = b.empid) as "Date employee was hired",

b.dob as "birth date",

trunc(((sysdate - b.dob)/365.25),0) as "age",

br.phoneext as "branch phone extension",

(select NVL(e.branchphone, 'no number availible')

from branch e

where e.branchid = br.branchid) as "phone number",

(select m.b\_name

from branch m

where m.branchid = br.branchid) as "branch name",

b.degree as "as highest degree earned",

b.degreedate as "date earned degree"

from Bank\_employee b

left join EMP\_ANNUAL\_DATA a

on b.empid = a.empid

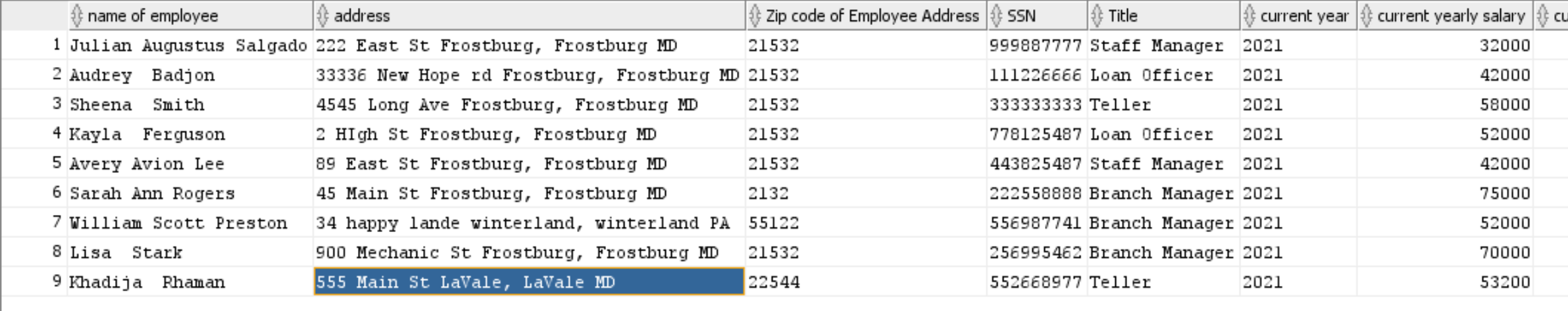
left join branch\_employee br

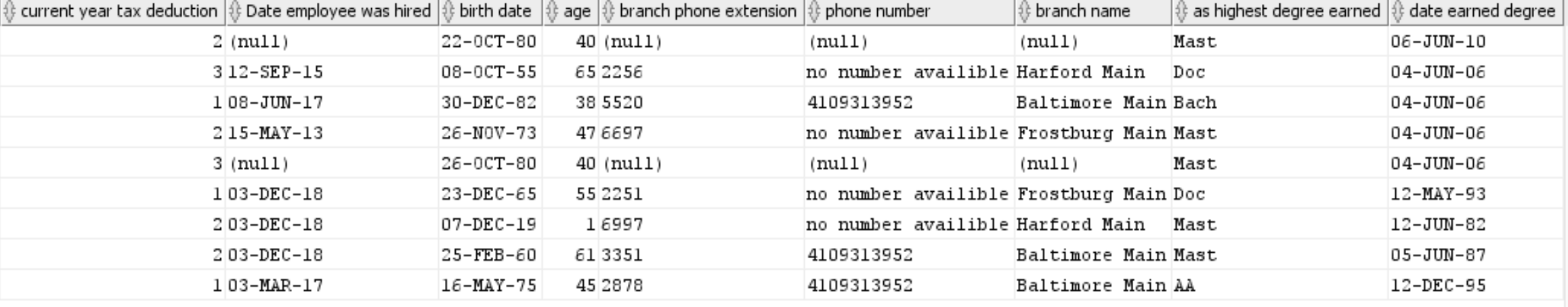
on br.empid = b.empid

where b.ssn is not null;

SELECT \*

FROM EMPLOYEE\_DATA;





1. **Employee\_salary with the following attributes:**

create view Employee\_salary as

select

distinct nvl(e.fname, '') || ' ' || nvl(e.mname, '') || ' ' || nvl(e.lname, '') as "name of employee",

to\_Char(sysdate, 'yyyy') as "current year",

e.ssn as "employee ssn",

(select c.salary

from EMP\_ANNUAL\_DATA c

where c.year <= to\_char(sysdate, 'YYYY')

and c.empid = e.empid

and rownum = 1) as "current salary",

nvl((select m.b\_name

from branch m

where m.branchid = b.branchid), 'no branch name availible') as "branch name",

nvl((select sum(ema.salary)

from EMP\_ANNUAL\_DATA ema

left join branch\_employee br

on ema.empid = br.empid

where br.branchid = (select x.branchid

from branch\_employee x

where e.empid = x.empid)),0)

as "total branch salaries",

nvl((select max(ema.salary)

from EMP\_ANNUAL\_DATA ema

left join branch\_employee br

on ema.empid = br.empid

where br.branchid = (select x.branchid

from branch\_employee x

where e.empid = x.empid)),0)

as "highest branch salary",

trunc(nvl((select avg(ema.salary)

from EMP\_ANNUAL\_DATA ema

left join branch\_employee br

on ema.empid = br.empid

where br.branchid = (select x.branchid

from branch\_employee x

where e.empid = x.empid)),0), 2)

as "average branch salary"

from bank\_employee e

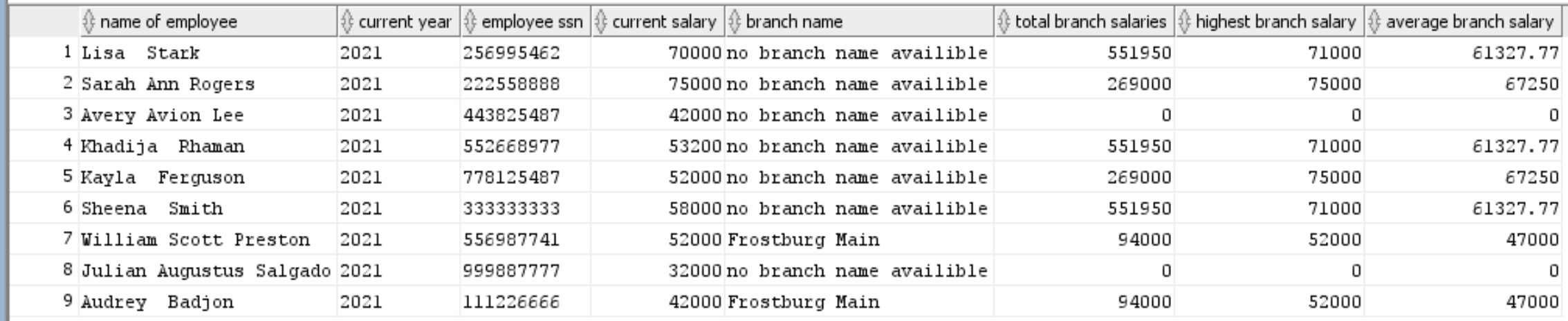
left join branch\_employee b

on e.empid = b.empid

left join branch a

on b.branchid = a.branchid;

select \* from Employee\_salary;



1. **Branch\_data with the following attributes:**

alter table branch

add fax\_number number;

create or replace view Branch\_data as

select

br.branchid,

br.b\_name as "branch name",

br.b\_st || ' ' || br.b\_city || ', ' || br.b\_state || ', ' || br.b\_zip as "branch address",

nvl(br.branchphone, 'no phone listed') as "branch phone",

br.fax\_number as "branch fax number",

(select fax\_number from dual) as "fax number",

(select count(\*)

from branch\_employee be

where br.branchid = be.branchid) as "number of branch employees",

br.category,

be.fname || ' ' || be.lname as "brnach manager",

(select count(\*)

from DEPOSIT\_ACCT\_TRANSACTION dat

where dat.accessptid = bap.accesspointid) as "number of branch transactions"

from branch br

left join branch\_manager bm

on bm.branchid =br.branchid

left join bank\_employee be

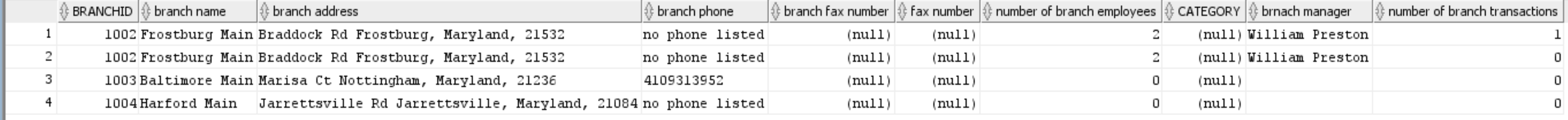
on bm.empid = be.empid

left join branch\_access\_points bap

on br.branchid = bap.branchid;

select \*

from Branch\_data;



1. **Valued\_Customers with the following attributes:**

alter table bank\_customer

add email varchar2(512);

create or replace view Valued\_Customers as

select distinct

bc.ssn,

bc.fname || ' ' || bc.lname as "name",

trunc(months\_between(sysdate, bc.dob)/12, 0) as "age",

bc.homephone as "home phone",

bc.workphone as "work phone",

bc.street || ' ' || bc.city as "address",

bc.zip,

bc.email,

bc.state,

(select count(cat.amount)

from credit\_acct\_transaction cat

where to\_char(cast(cat.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1

and cat.creditacctno = ca.creditacctno)

+

(select count(dt.amount)

from DEPOSIT\_ACCT\_TRANSACTION dt

where to\_char(cast(dt.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1

and ltrim(dt.acctno, 'CDEP0') = ltrim(ca.creditacctno, 'CCR0')) as "Total transactions",

nvl((select sum(dt.amount)

from deposit\_acct\_transaction dt

where to\_char(cast(dt.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1

and ltrim(dt.acctno, 'CDEP0') = ltrim(ca.creditacctno, 'CCR0')),0)

+

nvl((select sum(cat.amount)

from credit\_acct\_transaction cat

where cat.creditacctno = ca.creditacctno

and to\_char(cast(cat.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1),0)

as "Total transaction amount"

from bank\_customer bc

left join credit\_account ca

on bc.custid = ca.primary

left join DEPOSIT\_ACCt da

on bc.custid =da.primary

where nvl((select sum(dt.amount)

from deposit\_acct\_transaction dt

where to\_char(cast(dt.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1

and ltrim(dt.acctno, 'CDEP0') = ltrim(ca.creditacctno, 'CCR0')),0)

+

nvl((select sum(cat.amount)

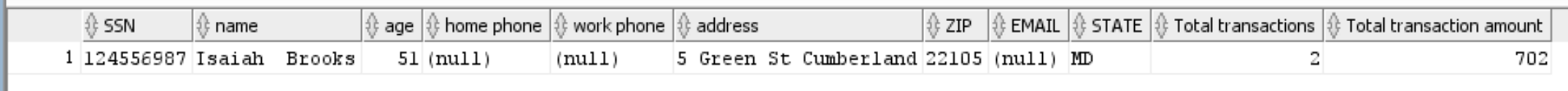
from credit\_acct\_transaction cat

where cat.creditacctno = ca.creditacctno

and to\_char(cast(cat.transdatetime as date), 'YYYY') = to\_char(sysdate, 'YYYY')-1),0) > 500;

select \* from valued\_customers;

I chose to use the total amount of all transactions as with bitcoin the main way miner will make money after all of the bitcoins are mined are on transactional fees.



1. **Statistics\_by\_Branch with the following attributes (Read only view):**

create or replace view Statistics\_by\_Branch as

select

b.branchid,

b.b\_name,

to\_char(sysdate, 'yyyy') as "year",

(select count(\*)

from DEPOSIT\_ACCT\_TRANSACTION da

left join branch\_access\_points bap

on da.accessptid = bap.accesspointid

where bap.branchid = b.branchid) as "deposits at branch",

((select count(\*)

from credit\_acct\_transaction cat)

+

(select count(\*)

from DEPOSIT\_ACCT\_TRANSACTION dt

)) as "total transactions",

(select count(\*)

from bank\_employee be

left join branch\_employee br

on be.empid =br.empid

where br.branchid = b.branchid) as "total employees at branch"

from branch b;

select \* from Statistics\_by\_Branch;

