Quiz 1 Review
Quiz 1 Review

• ER Diagrams
• SQL Queries
• Postgres copy statements
• Stored Procedures
• Triggers
Library ER Diagram

- **Library**
  - Libnum
  - Capacity

- **Staff**
  - Sid
  - First
  - Last

- **WorksAt**
  - (0, M)

- **Houses**
  - (0, M)

- **Copy**
  - Copynum
  - Price
  - (1, 1)

- **Managed**

- **Has**
  - (1, 1)
  - (0, M)

- **Book**
  - Booknum
  - Title
  - Pages
  - (0, M)

- **Author**
  - Aid
  - First
  - Last
  - (0, M)

- **Percent**
  - (0, M)

- **Written**

Relationships:
- (1, 1) between Staff and Managed
- (0, M) between WorkAt and Houses
- (0, M) between Houses and Managed
- (1, 1) between Managed and Copy
- (1, 1) between Copy and Booknum
- (0, M) between Book and Title
- (0, M) between Book and Pages
- (0, M) between Author and First
- (0, M) between Author and Last
- (0, M) between Percent and Written
Create Tables

- CREATE TABLE Staff(sid serial primary key, first VARCHAR(30), last VARCHAR(30), libnum int not null, mid int, foreign key (libnum) references Library(libnum), foreign key (mid) references Staff(sid))
### Meta Data

```sql
cs464lib2=# \d staff

Table "public.staff"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid</td>
<td>integer</td>
<td>not null</td>
</tr>
<tr>
<td></td>
<td></td>
<td>default</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nextval('staff_sid_seq'::regclass)</td>
</tr>
<tr>
<td>first</td>
<td>character varying(30)</td>
<td></td>
</tr>
<tr>
<td>last</td>
<td>character varying(30)</td>
<td></td>
</tr>
<tr>
<td>libnum</td>
<td>integer</td>
<td>not null</td>
</tr>
<tr>
<td>mid</td>
<td>integer</td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "staff_pkey" PRIMARY KEY, btree (sid)

Foreign-key constraints:
- "staff_libnum_fkey" FOREIGN KEY (libnum) REFERENCES library(libnum)
- "staff_mid_fkey" FOREIGN KEY (mid) REFERENCES staff(sid)

Referenced by:
- TABLE "staff" CONSTRAINT "staff_mid_fkey" FOREIGN KEY (mid) REFERENCES staff(sid)
```
Insert Data into Staff

• Load data with sids from file
  – copy staff(sid,first,last,libnum,mid) from '/Users/gendreau/classes/spring17/cs464/SQLReview/staff.txt' with null '*\' delimiter '|';

• Reset starting sid for automatic assignment of sids
  – alter sequence staff_sid_seq restart 29;

• Insert a new row into the Staff table
  – Insert into Staff(first,last,libnum,mid) values ('Jane', 'Doe', 2, 16)

• Load data without sids from file
  – copy staff(first,last,libnum,mid) from '/Users/gendreau/classes/spring17/cs464/SQLReview/staff2.txt' with null '*\' delimiter '|';
Example Query problems

• Find the sid, first and last for staff who are managed by staff member 2
• Find sid, first and last for staff who have no manager
• Find the number of staff members managed by each staff member who manages at least one staff member. The result should be tuples of the form (sid, first, last, numEmployees);
• Find the library with the largest number of staff members
• Find libraries that have at least 10 staff members
• Find sid, first and last of staff who work in a library that has a copy of all books written by Anton Checkhov
More Tables

CREATE TABLE Customer(cid int primary key, first VARCHAR(30), last VARCHAR(30));

CREATE TABLE Loan(loannum serial primary key, out date not null, due date not null, ret date, fine money default 0.0, paid boolean default true, copynum int not null, sid int not null, cid int not null, foreign key (copynum) references Copy(copynum),foreign key (sid) references Staff(sid), foreign key (cid) references Customer(cid));

create table LibLog(logId serial primary key, oldCapacity integer, newCapacity integer, changeDate Date, libnum integer references Library(libnum));
create or replace function insertcopy(bnum integer, name text, length integer, cost money, lib integer) returns integer AS $$

Declare
    bknum integer;
    cnum integer;
Begin
    select booknum into bknum from Book where booknum = bnum;
    if bknum is NULL then
        insert into Book (booknum, title, pages) values (bnum, name, length);
    end if;

    insert into copy (booknum, price, libnum) values (bnum, cost, lib)
    returning copynum into cnum;

    return cnum;
End;
$$ language plpgsql;
create or replace function makeLoan(cus integer, cpnum integer, snum integer)
returns integer AS $$
Declare
    sLib integer;
    cpLib integer;
    lnunm integer;

Begin
    select libnum into sLib from staff where sid = snum;
    select libnum into cpLib from copy where copynum = cpnum;
    if sLib = cpLib then
        insert into Loan (out, due, copynum, sid, cid) values (current_date, current_date+21, cpnum, snum, cus)
        returning loannum into lnunm;
    else
        lnunm = NULL;
    end if;

return lnunm;

End;
$$ Language plpgsql;
numEmployees

create or replace function numEmployees(manager integer)
returns integer AS
$$
Declare
    num integer;
    empId integer;
Begin

    num := 0;
    for empID in select sid from Staff where mid = manager loop
        num = 1 + numEmployees(empID) + num;
    end loop;

    return num;
End;
$$ Language plpgsql;
create or replace function logCapChange() returns trigger AS $$
Declare

Begin
    insert into LibLog (oldCapacity, newCapacity, changeDate, libnum)
    values (OLD.capacity, NEW.capacity, current_date, NEW.libnum);

    return NEW;
END;
$$ Language plpgsql;

create trigger capChange before update of capacity on Library
    for each row execute procedure logCapChange();
Trigger Example

create or replace function checkFine() returns trigger AS $$
Declare
  late integer;
Begin
  if OLD.ret is NULL then
    late = current_date - OLD.due;
    if late > 0 then
      NEW.fine = late * 0.25;
      NEW.paid = false;
    end if;
  else
    raise exception '% previously returned', NEW.loannum;
  end if;

  return NEW;
End;
$$ Language plpgsql;
create trigger returnCopy before update of ret on loan
  for each row execute procedure checkFine();
space_available

create or replace function space_available(lib integer)
returns integer AS $$
Declare
    cap integer;
    used integer;
    avail integer;
Begin
    select L.capacity into cap
    from library L
    where L.libnum = lib;

    select count(copynum) into used
    from copy C
    where C.libnum = lib;

    avail := cap - used;
    return avail;
End;
$$ Language plpgsql;
create or replace function space_available2(lib integer, avail OUT integer) AS $$
Declare
  cap integer;
  used integer;
Begin
  select L.capacity into cap
  from library L
  where L.libnum = lib;

  select count(copynum) into used
  from copy C
  where C.libnum = lib;

  avail := cap - used;
End;
$$ Language plpgsql;
import java.sql.*;

public class FirstDB
{
    public static void main(String args[])
    {
        Connection con;
        try {
            con = DriverManager.getConnection("jdbc:postgresql://localhost/cs464lib", user, password);
            System.out.println("Connection Successful");
            Statement s = con.createStatement();
            ResultSet r = s.executeQuery("select * from author");
            while (r.next()) {
                System.out.println(r.getString(2) + " " + r.getString(3));
            }
        }
    }
}
CallableStatement s = con.prepareCall("{? = call space_available(?)}");
s.registerOutParameter(1, Types.INTEGER);
s.setInt(2, 2);
s.execute();
System.out.println(s.getInt(1));

CallableStatement s2 = con.prepareCall("{call space_available2(? , ?)}");
s2.registerOutParameter(2, Types.INTEGER);
s2.setInt(1, 2);
s2.execute();
s2.execute();
System.out.println(s2.getInt(2));
Java/JDBC

catch (Exception e)
{
    System.out.println("Connection failed"+e.getMessage());
}
}