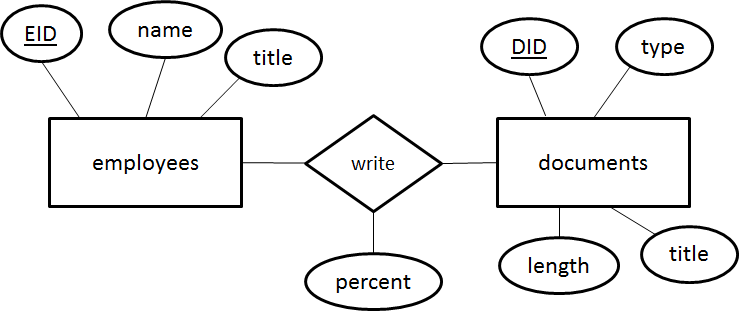
**CS 7330**

**Homework 7.1 MLO 7.2, 7.3, 7.4**

You will write and implement SQL commands.   
You need to identify the system used, provide a listing of the commands, and the resulting output. You can use screenshots, as well as trimming extraneous lines from the SQL output. If you submit your solution electronically, make sure it is organized as a single file.

For the following questions, I will be using <https://sqliteonline.com/> to store my tables and run the queries.

1) **Implement** a solution to the following relational schema



An employee can write more than one type of document (memo, internal-report, external-report, presentation) in this business. The percent of the Writes relation shows the percentage that the employee contributed to a given document.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EID | Name | title | DID | type | Length | title | percent |
| 100 | John Doe | CEO | 1 | external-report | 60 | 2015 Annual Report | 100 |
| 100 | John Doe | CEO | 2 | memo | 1 | 2015 Goals | 50 |
| 111 | Sally Financial | CFO | 2 | memo | 1 | 2015 Goals | 25 |
| 122 | HR Office | HR | 2 | memo | 1 | 2015 Goals | 25 |
| 100 | John Doe | CEO | 3 | external-report | 2 | 2015 OSHA Report | 50 |
| 133 | T Produce | Mgr | 3 | external-report | 2 | 2015 OSHA Report | 50 |
| 133 | T Produce | Mgr | 4 | internal-report | 5 | preliminary OSHA response | 100 |
| 133 | T Produce | Mgr | 5 | external-report | 10 | Gov OSHA response | 100 |
| 111 | Sally Financial | CFO | 6 | internal-report | 25 | Draft financial report | 100 |
| 111 | Sally Financial | CFO | 7 | external-report | 10 | Gov Req CFO report | 100 |
| 144 | Sam Shipper | Mgr | 8 | memo | 2 | Shipping requirements | 100 |

Creating the tables:

*-- Question 1*

CREATE TABLE employee

(

  EID INT NOT NULL,

  name VARCHAR(100) NOT NULL,

  title VARCHAR(100) NOT NULL,

  PRIMARY KEY (EID)

);

CREATE TABLE document

(

  DID INT NOT NULL,

  type VARCHAR(100) NOT NULL,

  length INT NOT NULL,

  title VARCHAR(100) NOT NULL,

  percentage int NOT NULL,

  EID INT NOT NULL,

  PRIMARY KEY (DID, EID),

  FOREIGN KEY (EID) REFERENCES employee(EID)

);

Inserting the data:

insert into employee values(100, 'John Doe', 'CEO');

insert into employee values(111, 'Sally Financial', 'CFO');

insert into employee values(122, 'HR Office', 'HR');

insert into employee values(133, 'T Produce', 'Mgr');

insert into employee values(144, 'Sam Shipper', 'Mgr');

insert into document values(1, 'external-report', 60, '2015 Annual Report', 100, 100);

insert into document values(2, 'memo', 1, '2015 Goals', 50, 100);

insert into document values(2, 'memo', 1, '2015 Goals', 25, 111);

insert into document values(2, 'memo', 1, '2015 Goals', 25, 122);

insert into document values(3, 'external-report', 2, '2015 OHSA Report', 50, 100);

insert into document values(3, 'external-report', 2, '2015 OHSA Report', 50, 133);

insert into document values(4, 'internal-report', 5, 'Prelim OHSA Response', 100, 133);

insert into document values(5, 'external-report', 10, 'Gov OHSA Response', 100, 133);

insert into document values(6, 'internal-report', 25, 'Draft financial report', 100, 111);

insert into document values(7, 'external-report', 10, 'Gov req CFO report', 100, 111);

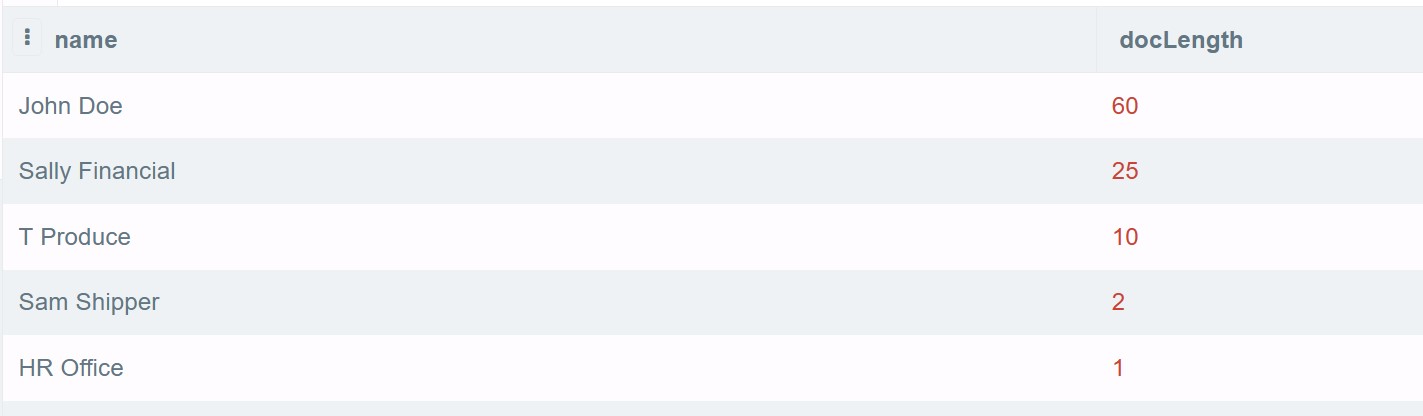
insert into document values(8, 'memo', 2, 'Shipping requirements', 100, 144);

a. Print the names of the longest report. You can assume there is only one longest report.

select e.name, MAX(d.length) docLength from employee e

left join document d on d.EID = e.EID

group by e.name

order by docLength DESC;

b. Print the names and titles of all employees who have written 100% of at least one internal-report and at least one external report

select e.name, e.title from employee e

where e.eid in

(

  select d.EID from document d where d.type = 'external-report' AND d.percentage = 100

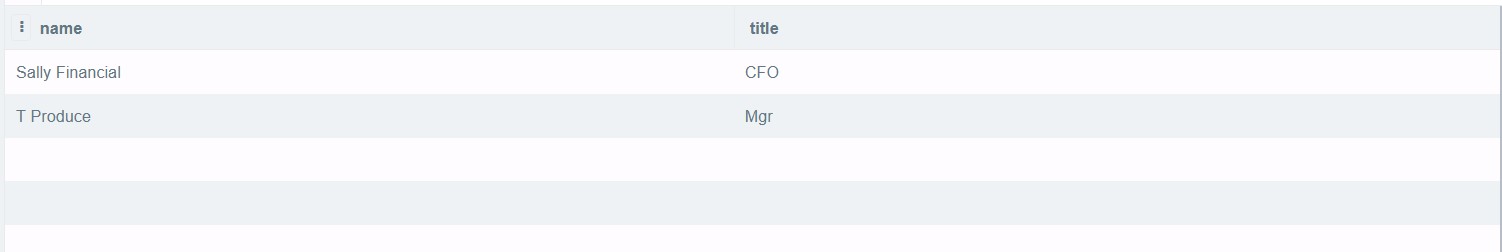
)

and e.eid in

(

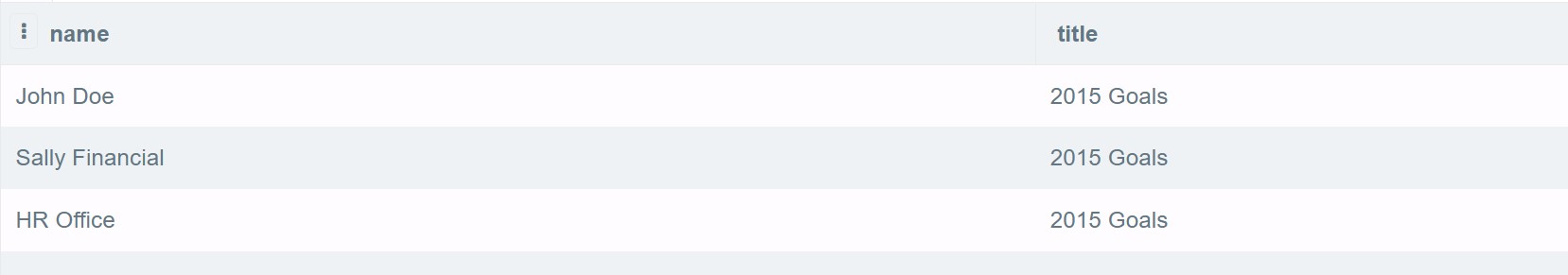
    select d.eid from document d where d.type = 'internal-report' AND d.percentage = 100

);

  
c. List the document title and all authors, of the document that had the most authors.

select e.name, d.title from employee e

join document d on e.EID = d.EID

where d.DID IN

d. Which job title of employee is associated with the most documents by effort?

select TOP 1 e.title, SUM(d.percentage) SumEffort from employee e

    join document d on d.EID = e.EID

    GROUP by e.title

    ORDER by SumEffort desc;

e. Find the names of all employees who have written the least amount of documents.

select e.name, COUNT(d.did) works from employee e

join document d on d.EID = e.EID

GROUP by e.name

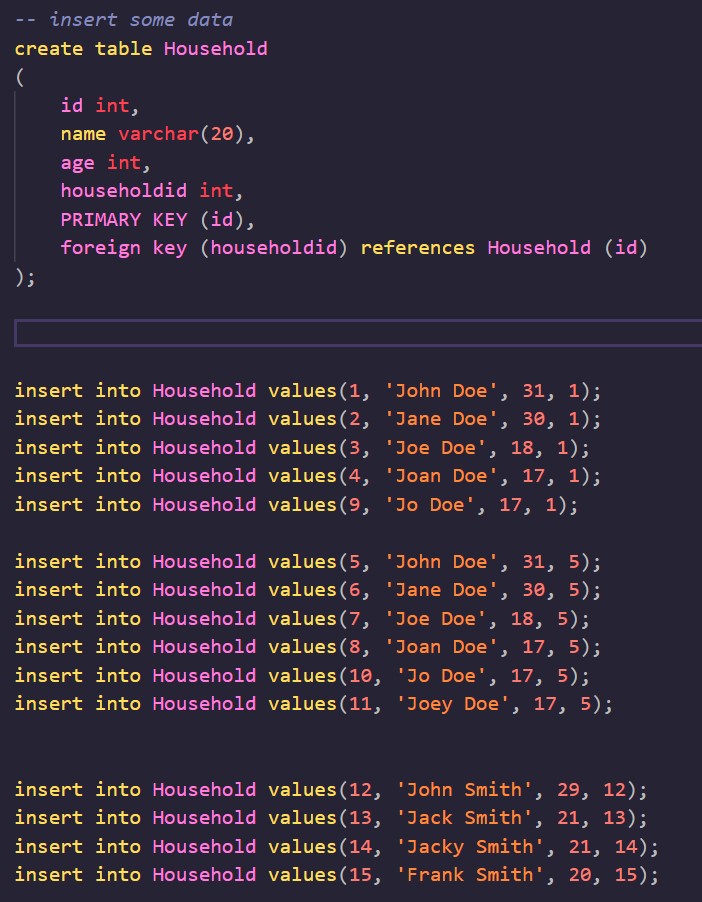
order by works ASC;

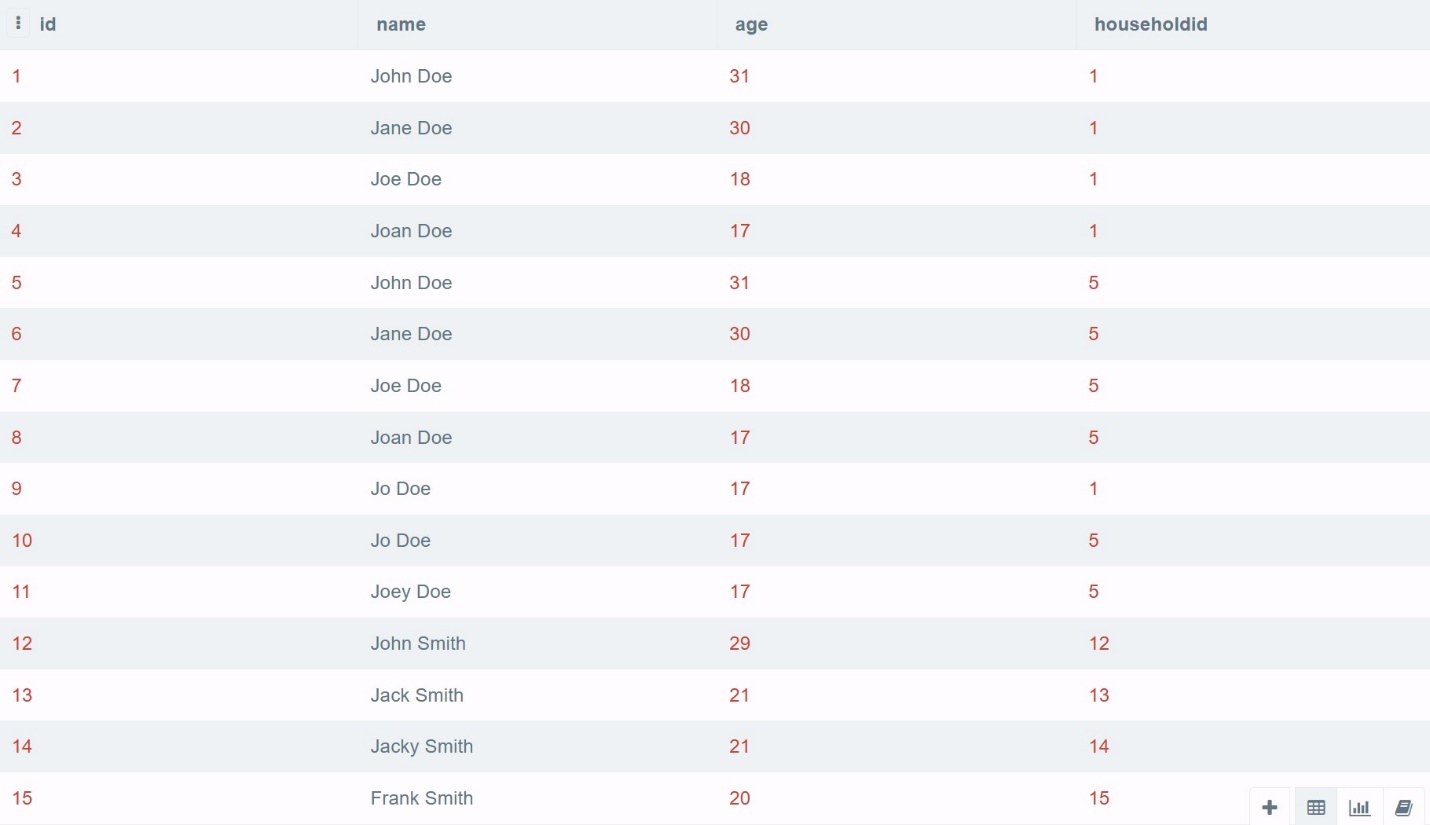


2. Given the relation: Households ( id, name, age, householdid). For any tuple, if householdid = id, then this tuple is the “head” of the household, otherwise householdid identifies the household that id belongs to.

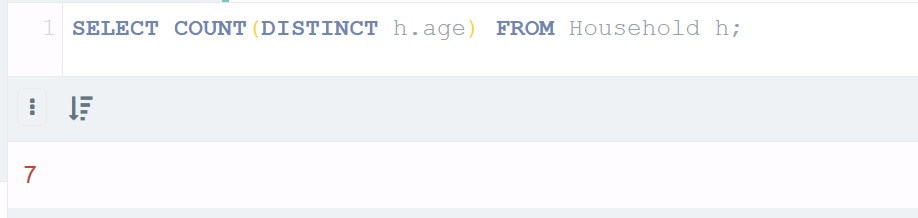
Implement your solution with data to answer.....

1. Make sure the table structure enforces that householdid is an existing id. Show the table creation and data insert.





1. How many unique ages are there among all households?

select COUNT(DISTINCT h.age) from Household h;

1. List the names and ages of the largest household.

SELECT h.name, h.age, h.householdid from Household h

where h.householdid in (select TOP 1 h.householdid from Household h

GROUP by h.householdid

ORDER by COUNT(\*) desc);



1. What is the count of single member households?

select COUNT(\*) from

(

    select h.householdid from Household h

    group by h.householdid

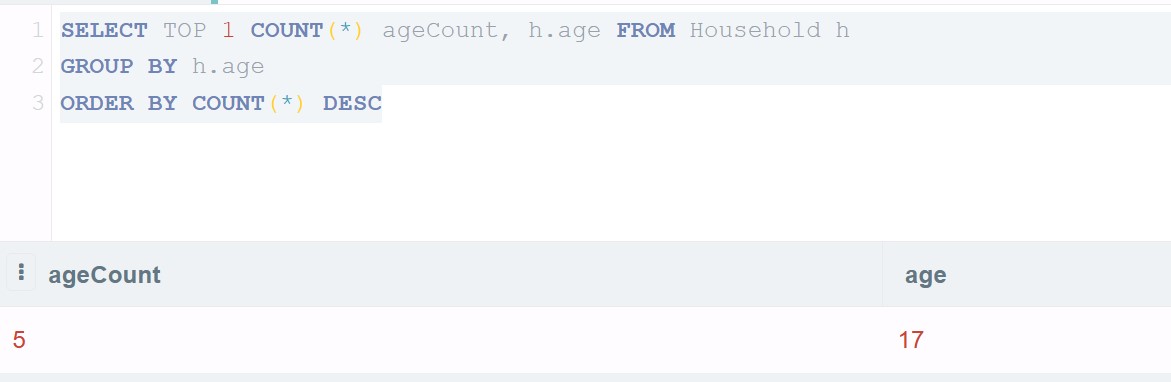
    HAVING count(\*) = 1

) as foo;

1. How many people are in the largest group of people of the same age? What is the age and the count?

select TOP 1 COUNT(\*) ageCount, h.age from Household h

group by h.age

ORDER by COUNT(\*) DESC

(END)

# Reference

Here is the whole file for the queries

*-- Question 1*

CREATE TABLE employee

(

  EID INT NOT NULL,

  name VARCHAR(100) NOT NULL,

  title VARCHAR(100) NOT NULL,

  PRIMARY KEY (EID)

);

CREATE TABLE document

(

  DID INT NOT NULL,

  type VARCHAR(100) NOT NULL,

  length INT NOT NULL,

  title VARCHAR(100) NOT NULL,

  percentage int NOT NULL,

  EID INT NOT NULL,

  PRIMARY KEY (DID, EID),

  FOREIGN KEY (EID) REFERENCES employee(EID)

);

*--insert HW data entries*

insert into employee values(100, 'John Doe', 'CEO');

insert into employee values(111, 'Sally Financial', 'CFO');

insert into employee values(122, 'HR Office', 'HR');

insert into employee values(133, 'T Produce', 'Mgr');

insert into employee values(144, 'Sam Shipper', 'Mgr');

insert into document values(1, 'external-report', 60, '2015 Annual Report', 100, 100);

insert into document values(2, 'memo', 1, '2015 Goals', 50, 100);

insert into document values(2, 'memo', 1, '2015 Goals', 25, 111);

insert into document values(2, 'memo', 1, '2015 Goals', 25, 122);

insert into document values(3, 'external-report', 2, '2015 OHSA Report', 50, 100);

insert into document values(3, 'external-report', 2, '2015 OHSA Report', 50, 133);

insert into document values(4, 'internal-report', 5, 'Prelim OHSA Response', 100, 133);

insert into document values(5, 'external-report', 10, 'Gov OHSA Response', 100, 133);

insert into document values(6, 'internal-report', 25, 'Draft financial report', 100, 111);

insert into document values(7, 'external-report', 10, 'Gov req CFO report', 100, 111);

insert into document values(8, 'memo', 2, 'Shipping requirements', 100, 144);

*-- 1a*

select e.name, MAX(d.length) docLength from employee e

left join document d on d.EID = e.EID

group by e.name

order by docLength DESC;

*-- 1b*

*-- gets the reports that have 100% participation in either external-reports or internal-reports*

select e.name, e.title from employee e

where e.eid in

(

  select d.EID from document d where d.type = 'external-report' AND d.percentage = 100

)

and e.eid in

(

    select d.eid from document d where d.type = 'internal-report' AND d.percentage = 100

);

*-- 1c*

*--gets how many authors there are on a document*

select e.name, d.title from employee e

join document d on e.EID = d.EID

where d.DID IN

(

*--gets document that is most worked on*

    select TOP 1 d.did

    from document d

    GROUP by d.did

    order by COUNT(d.did) desc

);

*-- 1d*

*-- who is the title that has the most percentage sum?*

select TOP 1 e.title, SUM(d.percentage) SumEffort from employee e

    join document d on d.EID = e.EID

    GROUP by e.title

    ORDER by SumEffort desc;

*-- 1e*

*-- lists who has written the least documents*

select e.name, COUNT(d.did) works from employee e

join document d on d.EID = e.EID

GROUP by e.name

order by works ASC;

*-- Question 2*

*-- insert some data*

create table Household

(

    id int,

    name varchar(20),

    age int,

    householdid int,

    PRIMARY KEY (id),

    foreign key (householdid) references Household (id)

);

insert into Household values(1, 'John Doe', 31, 1);

insert into Household values(2, 'Jane Doe', 30, 1);

insert into Household values(3, 'Joe Doe', 18, 1);

insert into Household values(4, 'Joan Doe', 17, 1);

insert into Household values(9, 'Jo Doe', 17, 1);

insert into Household values(5, 'John Doe', 31, 5);

insert into Household values(6, 'Jane Doe', 30, 5);

insert into Household values(7, 'Joe Doe', 18, 5);

insert into Household values(8, 'Joan Doe', 17, 5);

insert into Household values(10, 'Jo Doe', 17, 5);

insert into Household values(11, 'Joey Doe', 17, 5);

insert into Household values(12, 'John Smith', 29, 12);

insert into Household values(13, 'Jack Smith', 21, 13);

insert into Household values(14, 'Jacky Smith', 21, 14);

insert into Household values(15, 'Frank Smith', 20, 15);

*-- 2b*

select COUNT(DISTINCT h.age) from Household h;

*-- 2c*

SELECT h.name, h.age, h.householdid from Household h

where h.householdid in (select TOP 1 h.householdid from Household h

GROUP by h.householdid

ORDER by COUNT(\*) desc);

*-- 2d*

select COUNT(\*) from

(

    select h.householdid from Household h

    group by h.householdid

    HAVING count(\*) = 1

) as foo;

*--2e*

select TOP 1 COUNT(\*) ageCount, h.age from Household h

group by h.age

ORDER by COUNT(\*) DESC