

CpSc 4620/6620 Quiz #7

Name: _____ ID: _____

Question #1 (10 points): Given the following relational table “Person”,

Person:

Name	Age	Weight
Harry	34	80
Sally	28	64
George	29	70
Helena	54	54
Peter	34	80

List the results of $\pi_{\text{Name, Weight}}(\text{Person})$ and $\sigma_{\text{Weight} \geq 64}(\text{Person})$ respectively

$\pi_{\text{Name, Weight}}(\text{Person})$

Name	Weight
Harry	80
Sally	64
George	70
Helena	54
Peter	80

$\sigma_{\text{Weight} \geq 64}(\text{Person})$

Name	Age	Weight
Harry	34	80
Sally	28	64
George	29	70
Peter	34	80

Question #2 (25 points): Given the following two tables:

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales
Marry	1000	Human Resource

Dept

DeptName	Manager
Finance	George
Sales	Harriet
Production	Charles

Please answer the following questions:

1. List the results of the relational operation **Employee** ⋈ **Dept**.

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet

2. List the results of the relational operation **Employee** ⋉ **Dept**.

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

3. List the results of relational operation **Employee** ⋈_≠ **Dept**.

Name	EmpId	DeptName
Marry	1000	Human Resource

4. List the results of relational operation **Employee** =X **Dept**.

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet
Marry	1000	Human Resource	NULL

5. List the results of relational operation **Employee X= Dept.**

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet
NULL	NULL	Production	Charles

Question #3 (5 points): Given the following two tables, please list the results of the relational operation:

Car  Boat
CarPrice < BoatPrice

Car

CarModel	CarPrice
CarA	20'000
CarB	30'000
CarC	50'000

Boat

BoatModel	BoatPrice
Boat1	10'000
Boat2	40'000
Boat3	60'000

Answer:

CarModel	CarPrice	BoatModel	BoatPrice
CarA	20'000	Boat2	40'000
CarA	20'000	Boat3	60'000
CarB	30'000	Boat2	40'000
CarB	30'000	Boat3	60'000
CarC	50'000	Boat3	60'000

Question #4 (10 points): Given the following three tables, please list the results of the relational operation:

1) Completed \div DBProject

2) Completed \div Project1

Completed

Student	Task
Fred	Database1
Fred	Database2
Fred	Compiler1
Eugene	Database1
Eugene	Compiler1
Sara	Database1
Sara	Database2

DBProject

Task
Database1
Database2

Project1

Task
Database1
Compiler1

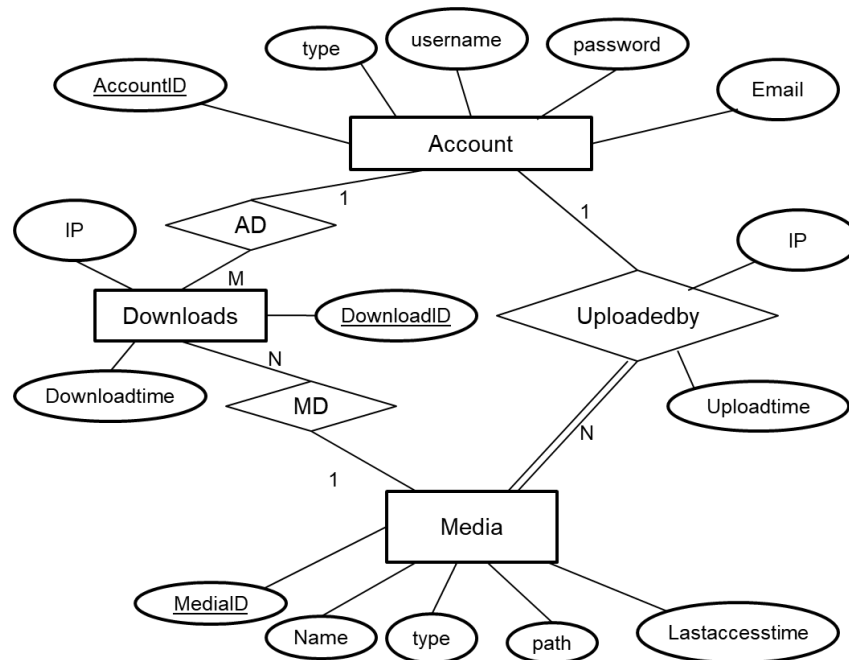
Completed ÷ DBProject

Student
Fred
Sara

Completed ÷ Project1

Student
Fred
Eugene

Question #5 (20 points): Given the following ER diagram:



Please answer the following questions:

1. Write MySQL statements to create the database schema **(5 points)**.

```

CREATE TABLE ACCOUNT (
    AccountID INTEGER AUTO_INCREMENT PRIMARY KEY ,
    email VARCHAR(1024) NOT NULL,
    username VARCHAR(100) NOT NULL,
    password VARCHAR(100) NOT NULL,
    type INTEGER NOT NULL
);
    
```

```

CREATE TABLE MEDIA (
    MediaID INTEGER AUTO_INCREMENT PRIMARY KEY,
    type INTEGER NOT NULL,
    name VARCHAR(100) NOT NULL,
    path VARCHAR (4096) NOT NULL,
    lastaccess DATETIME NOT NULL,
    AccountID INTEGER NOT NULL,
    uploadtime DATETIME NOT NULL,
    ip VARCHAR (16) NOT NULL,
    FOREIGN KEY(AccountID) REFERENCES ACCOUNT(AccountID)
);

```

```

CREATE TABLE DOWNLOADS (
    DownloadID INTEGER AUTO_INCREMENT PRIMARY KEY,
    AccountID INTEGER,
    MediaID INTEGER NOT NULL,
    ip VARCHAR(16) NOT NULL,
    downloadtime DATETIME NOT NULL,
    FOREIGN KEY(AccountID) REFERENCES ACCOUNT(AccountID),
    FOREIGN KEY(MediaID) REFERENCES MEDIA(MediaID)
);

```

2. Write SQL statement to retrieve information from this database **(15 points)**:

- 1) List the AccountID and the number of times that the user downloaded media 1.

```

SELECT AccountID, count( AccountID ) AS cnt
FROM DOWNLOADS
WHERE MediaID =1
GROUP BY AccountID

```

- 2) List the AccountID and the number of downloads for users that downloaded at least 5 times.

```

SELECT AccountID, count( AccountID ) AS cnt
FROM DOWNLOADS
GROUP BY AccountID
HAVING cnt > 4

```

- 3) List the name of the most recently downloaded media and its download time.

```

SELECT name, downloadtime
FROM MEDIA, DOWNLOADS
WHERE MEDIA.MediaID = DOWNLOADS.MediaID
ORDER BY downloadtime DESC LIMIT 1;

```

- 4) List the name of the most frequently downloaded media and the associated number of downloads.

```
SELECT name, count( * ) AS cnt
FROM DOWNLOADS, MEDIA WHERE DOWNLOADS.mediaID = MEDIA.mediaID
GROUP BY name
ORDER BY cnt DESC LIMIT 1;
```

- 5) List username who downloaded media 1 but not media 3:

Any of the following three queries:

```
SELECT username FROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID AND MediaID = 1
AND username NOT IN (SELECT username FROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID AND MediaID = 3);
```

```
SELECT UA.username FROM ( SELECT ACCOUNT.username,
DOWNLOADS.MediaID FROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID
AND MediaID =1 ) AS UA WHERE UA.username NOT IN ( SELECT username
FROM ACCOUNT, DOWNLOADS WHERE ACCOUNT.AccountID =
DOWNLOADS.AccountID AND MediaID =3);
```

```
SELECT UA.username
FROM (SELECT ACCOUNT.username FROM ACCOUNT, DOWNLOADS WHERE
ACCOUNT.AccountID = DOWNLOADS.AccountID AND MediaID =1) AS UA
WHERE NOT EXISTS
(SELECT ACCOUNT.username FROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID AND MediaID =3
AND UA.username = ACCOUNT.username);
```

Some wrong answers:

```
SELECT UA.username FROM ( SELECT ACCOUNT.username,
DOWNLOADS.MediaID FROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID
AND MediaID =1 ) AS UA WHERE NOT EXISTS ( SELECT username
FROM ACCOUNT, DOWNLOADS WHERE ACCOUNT.AccountID =
DOWNLOADS.AccountID AND MediaID =3);
```

```
SELECT ACCOUNT.usernameFROM ACCOUNT, DOWNLOADS
WHERE ACCOUNT.AccountID = DOWNLOADS.AccountID AND MediaID =1 AND
MediaID <> 3;
```

Question #6 (15 points): This set of questions refers to the PC Store schema described below. For each question, choose the answer that best describes what the SQL statement does.

Schema:

Product (maker, model, type)

PC (model, speed, ram, hd, rd, price)

Laptop (model, speed, ram, hd, screen, price)

Printer (model, color, type, price)

Sample data:

Product

maker	model	type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2001	laptop
B	2002	laptop
B	2003	laptop
B	1007	pc
C	1008	pc
C	2008	laptop
C	2009	laptop
C	3002	printer
C	3003	printer
C	3006	printer
D	1009	pc
D	1010	pc
D	1011	pc
D	2007	laptop
E	1012	pc
E	1013	pc
E	2010	laptop
F	3001	printer
F	3004	printer
G	3005	printer
H	3007	printer

Laptop

model	speed	ram	hd	screen	price
2001	700	64	5	12.1	1448
2002	800	96	10	15.1	2584
2003	850	64	10	15.1	2738
2004	550	32	5	12.1	999
2005	600	64	6	12.1	2399
2006	800	96	20	15.7	2999
2007	850	128	20	15	3099
2008	650	64	10	12.1	1249
2009	750	256	20	15.1	2599
2010	366	64	10	12.1	1499

PC

model	speed	ram	hd	rd	price
1001	700	64	10	48xCD	799
1002	1500	128	10	12xDVD	2499
1003	866	128	10	8xDVD	1999
1004	866	64	10	12xDVD	999
1005	1000	128	20	12xDVD	1499
1006	1300	256	40	16xDVD	2119
1007	1400	128	80	12xDVD	2299
1008	700	64	30	24xCD	999
1009	1200	128	80	16xDVD	1699
1010	750	64	30	40xCD	699
1011	1100	128	60	16xDVD	1299
1012	350	64	7	48xCD	799
1013	733	256	60	12xDVD	2499

Printer

model	color	type	price
3001	true	ink-jet	231
3002	true	ink-jet	267
3003	false	laser	390
3004	true	ink-jet	439
3005	true	bubble	200
3006	true	laser	1999
3007	false	laser	350

1) **SELECT maker FROM Product**

WHERE type = "laptop" AND maker IN (SELECT maker from Product WHERE type = "pc")

- A) Finds manufacturers that make either Laptops or PCs
- B) Finds manufacturers that make both Laptops and PCs
- C) Finds manufacturers that make either Laptops or PCs, but no printers
- D) Finds manufacturers that make both Laptops and PCs, but no printers

2) **SELECT model FROM Laptop**

WHERE speed > ANY (SELECT speed from PC)

- A) Finds the fastest laptop
- B) Finds the fastest PC
- C) Finds a PC faster than some laptop
- D) Finds a laptop faster than some PC

3) **SELECT model FROM Printer**

WHERE color = "true" OR TYPE = "ink-jet" AND price < 1000

(Note: In the absence of parentheses, the Boolean operator AND is evaluated before the Boolean operator OR)

- A) Finds color, ink-jet printers priced under 1000
- B) Finds ink-jet printers priced under 1000 or color printers of any type and price
- C) Finds color printers priced under 1000 or ink-jet printers of any price
- D) Finds all printers that are either color, ink-jet, or are priced under 1000

4) **SELECT PC.model, price FROM PC NATURAL JOIN Product**

WHERE speed > 900

GROUP BY maker

HAVING COUNT(*) >= 2

- A) Finds models and prices of all PCs having speeds > 900.
- B) Finds models and prices of all PCs made by a manufacturer that makes more than 1 model of PC.
- C) Finds models and prices of all PCs having a speed greater than 900 made by a manufacturer that makes more than 1 model of PC of any speed.
- D) Finds models and prices of all PCs made by a manufacturer that makes more than 1 model of PC with speeds > 900.

5) **SELECT DISTINCT P1.maker**

FROM Product P1, Product P2

WHERE P1.maker = P2.maker AND P1.model <> P2.model

- A) Finds manufacturers that make more than one product of any type.
- B) Finds manufacturers that make more than one PC and more than one Laptop and more than one Printer.
- C) Finds manufacturers that make more than one product of the same type.
- D) Finds manufacturers that make at least one printer and one PC and one laptop

Question #7 (15 points): We want to develop a database for a company.

- The company is organized into departments with each department located in one or more of the buildings rented by the company. Each department has a unique name and a budget. Each building has a size (that is, number of square foot of space) and a unique building name. We keep track of each employee's name and social security number (the social security number is guaranteed to be unique, but the name is not).
- An employee must work for one and only one department. Each department may have zero or more employees.
- Each department may be located in one building or spread over more than one building. A company building may have no departments using its space (before any department has had a chance to move into it), one department, or several departments using its space.
- For each department we keep track, for each building in which it is located, the number of employees of that department that are located in that building.

Design a relational schema for this database that minimizes the number of tables.

employees = (ssn: integer, ename: char[20], dname: char[20])

departments = (dname: char[20], budget: real)

buildings = (bname: char[20], bsize: integer)

locatedIn = (dname: char[20], bname: char[20], EmpNum: integer)