Name

CWID

Final Exam

May 2nd, 2024
3:30pm - 5:30pm

CS480 - Database Organization

Results

Please leave this empty!

1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | Sum

Instructions

- Try to answer all the questions using what you have learned in class. Keep hard questions until the end.
- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**
- The exam is closed book and closed notes!

Consider the following vehicle producer database schema and example instance:

part

pid	pname	weight
1	wheel	43
2	windshield	26
3	motor	140
4	trunk	450
5	roof	230

productpart

productname	pid	quantity
cabriolet	1	4
cabriolet	2	1
cabriolet	3	1
cabriolet	4	1
motorbike	1	2
motorbike	2	1
motorbike	3	1

partsupplier

sname	pid	price
XYZ	1	256
XYZ	2	580
Z-top	3	1200
Fedora	1	229
Fedora	4	649

supplier

sname	county
XYZ	USA
Z-top	Mexico
Fedora	Canada
Tetrum	Canada

Hints:

- Attributes with black background form the primary key of a relation (e.g., *sname* for relation *supplier*)
- The attribute *pid* of relation *productpart* is a foreign key to relation *part*
- The attribute *pid* of relation *partsupplier* is a foreign key to relation *part*
- The attribute *sname* of relation *partsupplier* is a foreign key to relation *supplier*
- All foreign keys have been created with the **CASCADE** option.

Part 1.1 Relational Algebra (Total: 20 Points)

The queries in this part should be written using the **set semantics** version of relational algebra.

Question 1.1.1 (5 Points)

Write a **relational algebra** expression that returns the *weight* of all parts needed to build a product called `cabriolet`.

Solution

$$\pi_{weight}(part \bowtie \sigma_{productname=cabriolet}(productpart))$$

alternatively based on the feedback I provided during the exam.

$$\gamma_{sum(weight);(}part \bowtie \sigma_{productname=cabriolet}(productpart))$$

$$\gamma_{sum(weight);(}\rho_{adjweight}(\pi_{weight \cdot quantity}(part \bowtie \sigma_{productname=cabriolet}(productpart))))$$

Question 1.1.2 (7 Points)

Write a **relational algebra** expression that returns the name (`pname`) of parts that weight at least 100 lb and are supplied by a supplier from Canada or Mexico.

Solution

$$\pi_{pname}(\sigma_{country=Mexico \vee country=Canada}(supplier) \bowtie partsupplier \bowtie \sigma_{weight \geq 100}(part))$$

Question 1.1.3 (8 Points)

Write a **relational algebra** expression that returns the total cost (price times quantity for all parts required to build a product) for building a `motorbike` using supplier `XYZ` as the only supplier (all parts are supplied by `XYZ`).

Solution

$$Q_{partcost} = \rho_{cost}(\pi_{price \cdot quantity}(\sigma_{productname=motorbike}(productpart) \bowtie \sigma_{sname=XYZ}(partsupplier))$$
$$Q = \gamma_{sum(cost)}(Q_{partcost})$$

Part 1.2 SQL - DDL (Total: 7 Points)

Question 1.2.1 (7 Points)

Write an **SQL statement** that creates a new relation `order` that records information about orders. For each order we record a unique id (`oid`), the part (`pid`) that we ordered, the ordered `quantity`, and the supplier which supplied the part (`sname`).

Solution

```
CREATE TABLE order (
  oid INT PRIMARY KEY,
  pid INT REFERENCES part,
  sname VARCHAR(80) REFERENCES supplier,
  quantity INT
);
```

Part 1.3 SQL - Queries (Total: 20 Points)

Question 1.3.1 (4 Points)

Write an **SQL query** that returns the number of suppliers per country.

Solution

```sql
SELECT country, count(*)
FROM supplier
GROUP BY country;
```

Question 1.3.2 (5 Points)

Write an **SQL query** that returns for each product (`productname`) the total number of parts required to build the product (take quantity into account).

Solution

```
SELECT productname, sum(quantity)
FROM productpart
GROUP BY productname;
```

Question 1.3.3 (5 Points)

Write an **SQL query** that returns the name, weight, and quantity of parts needed to build a product (`productname`) called `motorbike`.

Solution

```
SELECT pname, weight, quantity
FROM part NATURAL JOIN productpart
WHERE productname = 'motorbike';
```

Question 1.3.4 (7 Points)

Write an **SQL query** that returns for each part the lowest price for this part and the supplier(s) which supply the part at this price. Return the partname (`pname`), price, and supplier name (`sname`).

Solution

```
SELECT sname, pname, price
FROM part p NATURAL JOIN partsupplier s
WHERE price IN (SELECT min(ps.price)
                FROM partsupplier ps
                WHERE ps.pid = s.pid);
```

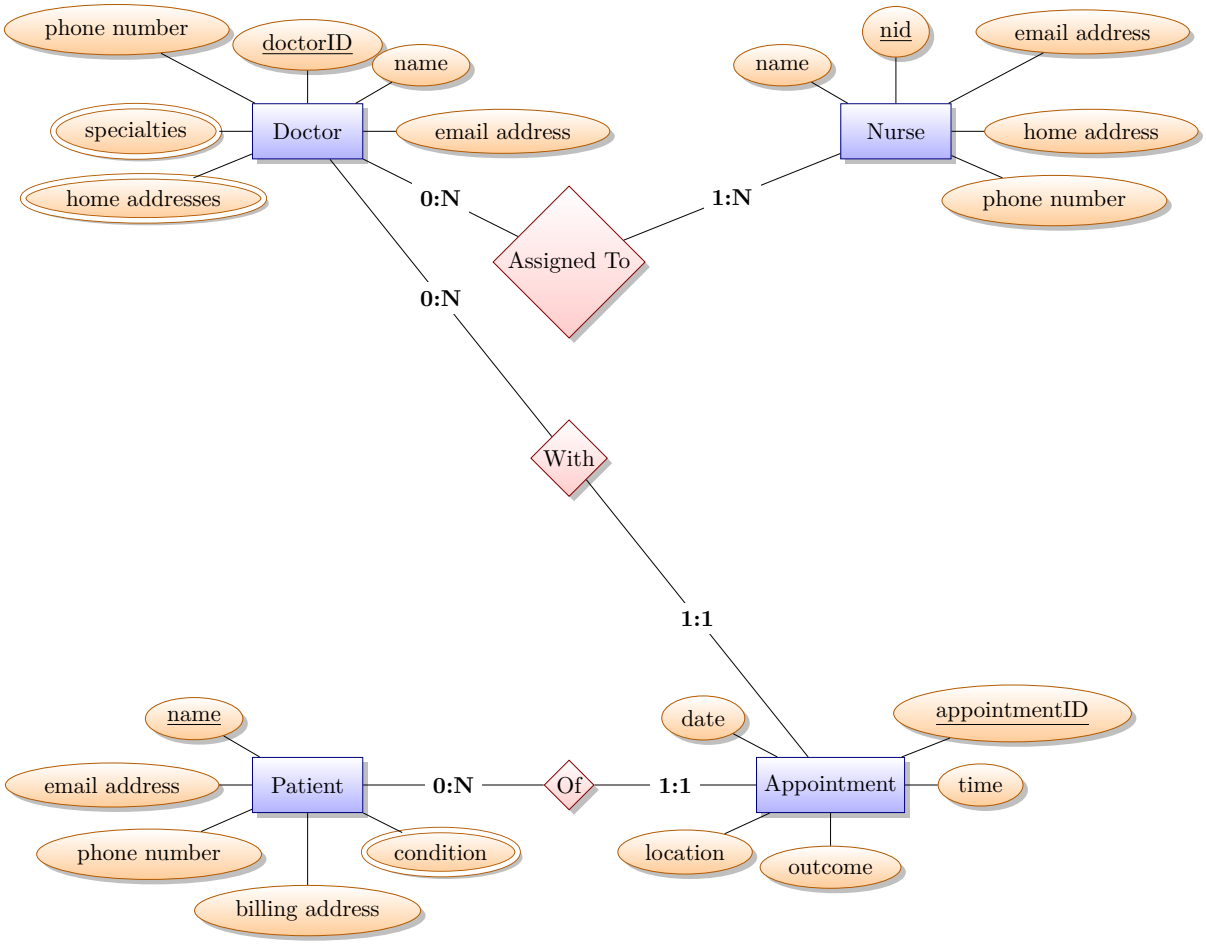
Part 1.4 ER Design (Total: 22 Points)

Question 1.4.1 (22 Points)

Design an ER diagram for a healthcare management system that includes doctors, nurses, patients, and appointments.

- **Doctor**: A doctor has a unique *doctorID*, *name*, *email address*, *phone number*, and one or more *home addresses*. Each doctor may have several *medical specialties* (e.g., cardiology or pediatrics).
- **Nurse**: A nurse has a *name*, *email address*, *phone number*, and *home address*. Each nurse is associated with one or more doctors.
- **Patient**: A patient has a unique *name*, *email address*, *phone number*, and *billing address*. Each patient may have one or more medical conditions (e.g., asthma or diabetes).
- **Appointment**: An appointment for a patient is with a specific doctor. The healthcare management system keeps track of the appointment and its associated *date*, *time*, *location*, and *outcome* (e.g., canceled or rescheduled).

Solution



- **Using any of the attributes of nurse as a key is ok as this was not specified**
- **Some cardinalities are not specified, so students were supposed to make educated guesses. The following are ok**
  - **Assigned to** for the doctor side we allow any cardinality
  - **With** for doctors either $0:N$ or $1:N$
  - **Of** for patient $0:N$ or $1:N$

Part 1.5 Normalization and Functional Dependencies (Total: 13 Points)

Consider the following relation $R(A, B, C, D, E, F)$ and functional dependencies Σ that hold over this relation.

$$\begin{aligned}\Sigma = \{ & \\ & F \to BC \\ & D \to A \\ & CD \to BF \\ & A \to E \\ & \}\end{aligned}$$

Question 1.5.1 (6 Points)

Determine all candidate keys of R.

Solution

$\{DF\}$ $\{CD\}$

Question 1.5.2 (7 Points)

Compute a canonical cover of Σ. Show each step of the generation according to the algorithm shown in class.

Solution

1th iteration: 1) Apply union rule to combine right-hand sides:

No change
1th iteration: 2) Find extraneous attribute:
B is redundant in the RHS of $CD \to BF$: replacing it with $CD \to F$

Canonical cover:

$$CD \to F$$
$$A \to E$$
$$D \to A$$
$$F \to BC$$

Part 1.6 Concurrency Control (Total: 18 Points)

Question 1.6.1 (18 Points)

For each of the following schedules determine which properties this schedule has. For example, a schedule may be *recoverable* and *cascade-less*. Consider the following notation for operations of transactions:

$w_1(A)$	transaction 1 wrote item A
$r_1(A)$	transaction 1 read item A
c_1	transaction 1 commits
a_1	transaction 1 aborts

$S_1 = w_1(A), r_2(A), w_2(B), r_3(B), w_1(B), r_4(B), c_1, c_2, c_3, c_4$
$S_2 = w_1(A), r_2(B), w_2(C), r_3(D), w_4(E), r_4(C), w_5(B), r_6(F), w_7(G), r_7(D), w_8(H), c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8$
$S_3 = r_1(A), r_1(B), w_1(C), r_2(A), w_2(B), w_2(D), c_2, w_3(D), c_3, r_4(B), w_4(E), c_4, r_5(E), c_5, c_1, c_3$

■ S_1 is recoverable

❑ S_1 is cascade-less

❑ S_1 is conflict-serializable

■ S_2 is recoverable

❑ S_2 is cascade-less

■ S_2 is conflict-serializable

■ S_3 is recoverable

■ S_3 is cascade-less

■ S_3 is conflict-serializable