DSCI551 Midterm II (Tuesday) - Results

Attempt 1 of 1

Written Mar 21, 2023 3:32 PM - Mar 21, 2023 5:20 PM

Your quiz has been submitted successfully.

Attempt Score **93.5 / 100 - A-**
Overall Grade (Highest Attempt) **93.5 / 100 - A-**

0. Note

All your codes answered in this exam are expected to run properly.

1. ER & Constraints, 35 points

Consider the following tables. Note the primary key of each table is underlined.
Purchase(buyer, seller, product, store, price)
Product(name, maker, category)
Laptop(name, OS) // note OS is operation system
Person(name, city, phone)

Sample data:
Purchase("John", "David", "iphone12", "store 1", 500.5)
Purchase("John", "Bill", "Macbook", "store 2", 2000)
Product("iphone12", "Apple", "Cell phone")
Product("Macbook", "Apple", "Laptop")
Laptop("Macbook", "macOS")
Person("John", "LA", 1234)
Person("David", "SFO", 4567)

Suppose buyer and seller are both foreign keys referring to the name of Person; product in Purchase is a foreign key referring to the name of Product. Note that a laptop is also a product.

Question 1 — 9 / 10 points

Reverse engineer the above tables into an ER model. State the entities, relationships, keys, and multiplicity of relationships.

Entities:
Person (name, city, phone)
Product (name, maker, category)
Laptop (name, OS)
Purchase (buyer, seller, product, store, price)

Keys:
Person: name
Product: name
Laptop: name
Purchase: buyer, seller, product, store

Relationships:
Laptop is a type of a product
Purchase is linked to buyer and seller who is a person
Purchase also linked to product which is a type of laptop

Multiplicity of relationships:
Person - one-to-many relationship with Purchase as a buyer/seller.
Product - one-to-many relationship with Purchase as a purchased product.
Laptop - one-to-one relationship with Product as a type of product.

▼ Hide question 1 feedback

Purchase is a relationship table and not an entity. many to many relationship between person and product

Question 2 **8.5 / 9 points**

Write a create table statement for all the tables. Be sure to properly define primary key and foreign keys for each table. Use cascade policy for all foreign keys on the operations that cause violations.

```
CREATE TABLE Person (
  name VARCHAR(50) PRIMARY KEY,
  city VARCHAR(50),
  phone VARCHAR(20)
);

CREATE TABLE Product (
  name VARCHAR(50) PRIMARY KEY,
  maker VARCHAR(50),
  category VARCHAR(50)
);

CREATE TABLE Laptop (
  name VARCHAR(50) PRIMARY KEY,
  OS VARCHAR(50),
  FOREIGN KEY (name) REFERENCES Product (name) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Purchase (
  buyer VARCHAR(50),
  seller VARCHAR(50),
  product VARCHAR(50),
  store VARCHAR(50),
  price DECIMAL(10,2),
  PRIMARY KEY (buyer, seller, product, store),
  FOREIGN KEY (buyer) REFERENCES Person (name) ON DELETE CASCADE,
  FOREIGN KEY (seller) REFERENCES Person (name) ON DELETE CASCADE,
  FOREIGN KEY (product) REFERENCES Product (name) ON DELETE CASCADE
);
```

▼ Hide question 2 feedback

missing on update cascade

Question 3 **4 / 8 points**

For each foreign key you defined in question 2, indicate if you can use the "set null" policy instead. Explain your answer.

For table Purchase - buyer and seller cannot be set null
- it cannot be null in purchase records as it has to deleted if the person is dropped

For table Laptop - name can be set to null
- to remove its association with any other product

▼ Hide question 3 feedback

For table Laptop - name cannot be set to null

Question 4 **7 / 8 points**

If the name of a product needs to be changed in the Product table, which tables should the database server check for possible violations of foreign key constraints? What actions should be taken, based on your

definitions of tables in question 2.

Database server should check for Purchase table for foreign key violations -> Product table's primary key is referenced by the Purchase table's product column.

Actions:
- Update the name value in Product table, changes PK of the affected record, and any FK referencing it becomes invalid.
- Update the product value in Purchase table that matches the new name value of the corresponding Product record

Since we are using cascade, it takes care of the updates to be made.

▼ Hide question 4 feedback

would also check in laptop table

2. Views, 25 points

Using the same tables in Section 1, answer the following questions.

Question 5 — 9 / 9 points

Create a view named LALaptop that finds the name and OS of laptops sold to people living in LA. Column names for the view should be 'name' and 'OS'.

```
CREATE VIEW LALaptop AS
SELECT Laptop.name, Laptop.OS
FROM Laptop
JOIN Product ON Laptop.name = Product.name
JOIN Purchase ON Product.name = Purchase.product
JOIN Person ON Purchase.buyer = Person.name
WHERE Person.city = 'LA';
```

Question 6 — 8 / 8 points

Write a SQL query that uses the view defined in question 5 to find out how many times people living in LA bought laptops that run Mac OS. Please only return the number and set the returned column's name as 'frequency'.

```
SELECT COUNT(*) as frequency
FROM LALaptop
WHERE OS = 'macOS';
```

Question 7 — 8 / 8 points

Explain the difference between virtual view and materialized view. What are the advantages and disadvantages of each view?

Virtual view - database object acting as window to underlying table
- doesn't store any data while it computed only on-demand when each time the view is queried, hence slow at runtime
- keeps the data up-to-date

Advantages:
- no data duplication
- less storage required
- is always up-to-date

Disadvantages:
- slow execution
- reduced load on underlying tables

Materialized view - database object that stores the result into a physical table
- it is precomputed hence fast at runtime
- has to updated periodically

Advantages:

- fast execution, improved query performance
-

Disadvantages:
- requires more storage
- there are chances of data duplication
- periodic data updates are required

3. XML & XPath, 20 points

Suppose data in Section 1 are now stored in an XML file.

3.1 XML File

Question 8 **5 / 5 points**

Use the following template, complete the XML file to store sample data shown in Section 1. Note a laptop is also stored in a product element which has an additional sub-element for OS. (PLEASE FOLLOW the structure given below; you can delete the '...' in the given text)

```
<data>
  <purchases>
    <purchase>
      <buyer>John</buyer>
      <seller>David</seller>
      <product>iphone12</product>
      <store>store 1</store>
      <price>500.5</price>
    </purchase>
    <purchase>
      <buyer>John</buyer>
      <seller>Bill</seller>
      <product>Macbook</product>
      <store>store 2</store>
      <price>2000</price>
    </purchase>
  </purchases>
  <products>
    <product>
      <name>iphone12</name>
      <maker>Apple</maker>
      <category>Cell phone</category>
    </product>
    <product>
      <name>Macbook</name>
      <maker>Apple</maker>
      <category>Laptop</category>
      <OS>macOS</OS>
    </product>
  </products>
  <persons>
    <person>
      <name>John</name>
      <city>LA</city>
      <phone>1234</phone>
    </person>
    <person>
      <name>David</name>
      <city>SFO</city>
      <phone>4567</phone>
    </person>
  </persons>
</data>
```

3.2 XPath Expression

Write an XPath expression to answer each of the following questions.

Question 9 — 4 / 4 points

Find all sellers John bought Iphone12 from. Output names of sellers (not elements).

```
//purchase[buyer="John" and product="iphone12"]/seller/text()
```

Question 10 — 4 / 4 points

Find all makers of laptops. Output actual makers (not elements).

```
//product[category="Laptop"]/maker/text()
```

Question 11 — 4 / 4 points

Find names of all people who live in LA and whose phone number has a digit 4. Output actual names (not elements).

```
//person[city="LA" and contains(phone, "4")]/name/text()
```

Question 12 — 3 / 3 points

Find makers of laptops running Windows OS. Output actual makes (not elements).

```
//product[category="Laptop" and OS="Windows"]/maker/text()
```

4. MongoDB, 20 points

Suppose data in Section 1 are now stored in MongoDB collections. Create a collection for each table, except for Laptop. Laptops should be stored in the Product collection, with an additional field for OS.

4.1 Insertion

Question 13 — 5 / 5 points

Write insert statements to insert sample data in Section 1 into collections.

```
db.createCollection("Purchase")
db.createCollection("Product")
db.createCollection("Person")

db.Product.insertMany([
 {
  name: "iphone12",
  maker: "Apple",
  category: "Cell phone"
 },
 {
  name: "Macbook",
  maker: "Apple",
  category: "Laptop",
  OS: "macOS"
 }
])

db.Person.insertMany([
 {
  name: "John",
  city: "LA",
  phone: "1234"
 },
 {
  name: "David",
  city: "SFO",
  phone: "4567"
 }
])
```

```
])

db.Purchase.insertMany([
 {
  buyer: "John",
  seller: "David",
  product: "iphone12",
  store: "store 1",
  price: 500.5
 },
 {
  buyer: "John",
  seller: "Bill",
  product: "Macbook",
  store: "store 2",
  price: 2000
 }
])
```

4.2 MongoDB Operation

Write a MongoDB operation to answer each of the following questions.

Question 14 — 3 / 3 points

Find all sellers John bought Iphone12 from. Output seller names only.

```
db.Purchase.aggregate([
 { $match: { buyer: "John", product: "iphone12" } },
 { $group: { _id: "$seller" } },
 { $project: { _id: 0, seller: "$_id" } }
])
```

Question 15 — 3 / 3 points

Find all makers of laptops. Output makers only.

```
db.Product.distinct("maker", { category: "Laptop" })
```

Question 16 — 3 / 3 points

Find names of all people who live in LA and whose phone number has a digit 4. Output names only.

```
db.Person.find(
 {
  city: "LA",
  phone: { $mod: [10, 4] }
 },
 {
  _id: 0,
  name: 1
 }
)
```

Question 17 — 3 / 3 points

Find out, for each maker, how many products for each category the maker has made.

```
db.Product.aggregate([
 { $unwind: "$category" },
 {
  $group: {
   _id: { maker: "$maker", category: "$category" },
   count: { $sum: 1 }
  }
 },
 {
  $group: {
```

```
    _id: "$_id.maker",
    categories: {
      $push: {
        category: "$_id.category",
        count: "$count"
      }
    }
  }
 },
 {
  $project: {
    _id: 0,
    maker: "$_id",
    categories: 1
  }
 }
])
```

Question 18 **3 / 3 points**

Find out how many makers make laptops running Windows OS.

```
db.Product.aggregate([
 {
  $match: {
    category: "Laptop",
    OS: "Windows"
  }
 },
 {
  $group: {
    _id: "$maker",
    count: { $sum: 1 }
  }
 },
 {
  $match: {
    count: { $gt: 0 }
  }
 },
 {
  $count: "num_makers"
 }
])
```

Done