CIS*3530 Data Base Systems and Concepts

Fall 2020

Instructor: Fangju Wang

Assignment 2 (20%)

Due: Monday Oct 26, 2020

Submission By: Mitchell Van Braeckel, Student ID: 1002297

The following is the database schema used in the questions:

Part(*P#*, PName, Producer, Year, Price)
Customer(*C#*, CName, City)
Supply(S#, P#, C#, Quantity, Amount, Date)

Note: when writing queries, please express dates in the format: YYYY-MM-DD

Note: refer to Part as *p*, refer to Customer as *c*, refer to Supply as *s*

Note: refer to P#, C#, S# as PNum, CNum, SNum, respectively

Note: P#, C#, S#, Year, Quantity are integers; PName, Producer, CName, City are strings; Price, Amount are doubles;

Note: Assume that if a Customer has not been Supplied with a Part, they are not in the s-table (ie. all C# in s-table refer to a Customer who has been supplied with at least 1 part referred to by the P#). Similarly, a Part that has not been Supplied to a Customer will not be in the s-table

Note: Assume an entry (row) in s-table has at least quantity 1+ (otherwise they aren't supplying anything and shouldn't exist)

NOTE: Assume that each CName+City pair is unique, so using DISTINCT will remove duplicates, but won't remove unique customers because there isn't more than one customer with the same name in a city. However, the way I set up Q5-8 to only be FROM c and using subquery, it "loops through" all C# without a problem about removing incorrect duplicates or displaying the same customer multiple times.

NOTE: nvm, email confirmation from Fangju, assume listing CName+City unique

NOTE: "since DATE" means that date or afterwards

NOTE: Lecture 4.1 SQL, slide 35 states: "Any attribute in SELECT must be in GROUP BY"

NOTE: Fangju says optimization doesn't matter for answers

---

1. Express each of the following queries in relational algebra AND relational calculus AND SQL (2%) each:

Query 1: List names and cities of all the customers.

- Relational Algebra: $\Pi_{\text{CName, City}}(c)$
- Relational Calculus: $\forall$ cx(c) { <cx.CName, cx.City> : True }
- SQL: SELECT CName, City FROM c

Query 2: List names and prices of those parts produced by Apple or Samsung in 2020.

- Relational Algebra: $\Pi_{\text{PName, Price}}(\sigma_{\text{(Producer="Apple"} \vee \text{Producer="Samsung")} \wedge \text{Year=2020}}(p))$
- Relational Calculus:
  $\forall$ px(p) { <px.PName, px.Price> :
  (px.Producer="Apple" $\vee$ px.Producer="Samsung") $\wedge$ px.Year=2020 }
- SQL:
  SELECT PName, Price FROM p
  WHERE (Producer='Apple' OR Producer='Samsung') AND Year=2020

Query 3: List supply dates, quantities, and amounts of all the Apple parts produced in 2020.

- Relational Algebra: $\Pi_{\text{Date, Quantity, Amount}}(\sigma_{\text{Producer="Apple"} \wedge \text{Year=2020}}(p) \bowtie s)$
- Relational Calculus:
  $\forall$ sx(s) { <sx.Date, sx.Quantity, sx.Amount> :
  $\exists$ px(p) (px.PNum=sx.PNum $\wedge$ px.Producer="Apple" $\wedge$ px.Year=2020) }
- SQL:
  SELECT Date, Quantity, Amount FROM s, p
  WHERE s.PNum=p.PNum AND Producer='Apple' AND Year=2020

Query 4: For each supply with a quantity over 1000 and after 2019-01-01, list the part name, customer names, and customer cities.

- Relational Algebra: $\Pi_{\text{PName, CName, City}}(\sigma_{\text{Quantity>1000} \wedge \text{Date>2019-01-01}}(s) \bowtie p \bowtie c)$
- Relational Calculus:
  $\forall$ px(p), $\forall$ cx(c) { <px.PName, cx.CName, cx.City> :
  $\exists$ sx(s) (
  sx.PNum=px.PNum $\wedge$ sx.CNum=cx.CNum $\wedge$
  sx.Quantity>1000 $\wedge$ sx.Date>2019-01-01) }
- SQL:
  SELECT PName, CName, City FROM c, s, p
  WHERE s.PNum=p.PNum AND s.CNum=c.CNum AND
  Quantity>1000 AND Date>2019-01-01

Query 5: List names and cities of the customers who bought one or more Apple products.

- Relational Algebra: $\Pi_{\text{CName, Clty}}(\sigma_{\text{Producer="Apple"}}(p) \bowtie s \bowtie c)$
- Relational Calculus:
  ∀ cx(c) { <cx.CName, cx.City> :
  ∃ px(p), ∃ sx(s) (
  px.PNum=sx.PNum ∧ sx.CNum=cx.CNum ∧ px.Producer="Apple") }
- SQL:

```
SELECT DISTINCT CName, City FROM c
WHERE EXISTS(
        SELECT *  FROM s, p
        WHERE p.PNum=s.PNum AND s.CNum=c.CNum AND Producer='Apple'
)
```

Query 6: List names and cities of the customers who did not buy any Apple product.

- Relational Algebra: $\Pi_{\text{CName, City}}(c - \Pi_{\text{CNum, CName, City}}(\sigma_{\text{Producer="Apple"}}(p) \bowtie s \bowtie c))$
- Relational Calculus:
  ∀ cx(c) { <cx.CName, cx.City> :
  ¬(∃ px(p), ∃ sx(s) (
  px.PNum=sx.PNum ∧ sx.CNum=cx.CNum ∧ px.Producer="Apple")) }
- SQL:

```
SELECT DISTINCT CName, City FROM c
WHERE NOT EXISTS(
        SELECT *  FROM s, p
        WHERE p.PNum=s.PNum AND s.CNum=c.CNum AND Producer='Apple'
)
```

Query 7: List names and cities of the customers who bought Apple products only.

- Relational Algebra:
  $\Pi_{\text{CName, City}}($
  $\Pi_{\text{CNum, CName, City}}(\sigma_{\text{Producer="Apple"}}(p) \bowtie s \bowtie c) - \Pi_{\text{CNum, CName, City}}(\sigma_{\text{Producer}\neq\text{"Apple"}}(p) \bowtie s \bowtie c)$
)
- Relational Calculus:
  ∀ cx(c) { <cx.CName, cx.City> :
  ∀ sx(s) (sx.CNum=cx.CNum
  ⇒ ∃ px(p) (px.PNum=sx.PNum ∧ px.Producer="Apple")) ∧
  ∃ sy(s) (sy.CNum=cx.CNum) }
- SQL:

```
SELECT DISTINCT CName, City FROM c
WHERE 'Apple'=ALL(
        SELECT Producer FROM s, p
        WHERE p.PNum=s.PNum AND s.CNum=c.CNum
) AND EXISTS(
        SELECT * FROM s, p
        WHERE p.PNum=s.PNum AND s.CNum=c.CNum
)
```

Query 8: List names and cities of the customers who bought all the Apple products.

- Relational Algebra: $\Pi_{\text{CName, City}}$ (($\sigma_{\text{Producer="Apple"}}(p) \bowtie s \bowtie c$) / $\Pi_{\text{PNum}}(\sigma_{\text{Producer="Apple"}}(p))$)
- Relational Calculus:
  ∀ cx(c) { <cx.CName, cx.City> :
  ∀ px(p) (px.Producer="Apple"
  ⇒ ∃ sx(s) (sx.CNum=cx.CNum ∧ sx.PNum=px.PNum)) }
- SQL:

```
SELECT DISTINCT CName, City FROM c
WHERE (
        SELECT COUNT(*) FROM p
        WHERE Producer='Apple'
) = (
        SELECT COUNT(DISTINCT p.PNum) FROM s, p
        WHERE p.PNum=s.PNum AND s.CNum=c.CNum AND Producer='Apple'
)
```

---

2. Express each of the following queries in SQL (1%) each):

Query 9: Find the average price of Apple products.

- SQL: SELECT AVG(Price) FROM p WHERE Producer='Apple'

Query 10: For each Apple product, list product number, product name, and total supply quantity since Jan 1, 2019.

- SQL:
  SELECT p.PNum, PName, SUM(Quantity) FROM s, p
  WHERE p.PNum=s.PNum AND Producer='Apple' AND Date>=2019-01-01
  GROUP BY PNum, PName

Query 11: For each Apple product supplied to Guelph in 2020, list product number, product name, total supply quantity, and total supply amount.

- SQL:
  SELECT p.PNum, PName, SUM(Quantity), SUM(Amount) FROM c, s, p
  WHERE p.PNum=s.PNum AND s.CNum=c.CNum AND
  Producer='Apple' AND City='Guelph' AND
  Date BETWEEN 2020-01-01 AND 2020-12-31
  GROUP BY PNum, PName

Query 12: For each Apple product supplied to more than 10 different customers in Guelph in 2020, list product number, product name, total supply quantity, and total supply amount. Sort the result by product name.

- SQL:
  SELECT p.PNum, PName, SUM(Quantity), SUM(Amount) FROM c, s, p
  WHERE p.PNum=s.PNum AND s.CNum=c.CNum AND
  Producer='Apple' AND City='Guelph' AND
  Date BETWEEN 2020-01-01 AND 2020-12-31
  GROUP BY PNum, PName
  HAVING COUNT(DISTINCT c.CNum) > 10
  ORDER BY PName