ISYS2120 Assignment 4 (Concepts) (sem1, 2024)

Due: Sunday 27 October, 11:59pm Sydney time
Value: 5%
This assignment is **individual work**.

Late work policy: (from unit outline) Late submissions for assignments will incur a penalty of 5% of the maximum awardable marks for each day, or part-day, past the due date, up to a maximum of 7 days (as after this time, feedback on on-time submissions will be available, resulting in an unfair advantage if submissions after this time were accepted). After 7 days late submissions will not be accepted. Where special consideration is granted for these assessments, extensions of a maximum of 7 days will be permitted. After 7 days, reweighting of other relevant tasks will be applied.

Summary (but details further down are authoritative)
Done individually. You are allowed to use GenerativeAI, but if so, you must submit an appendix showing each prompt you used, and the corresponding output from the computer; you will be marked only for what you have contributed, not on what was done by the computer.
Provided for you: four questions, and a template for the cover page
You produce and submit [on Canvas]: a pdf document, with cover page as shown in the template, and then your answers to the four questions.
Marks: these will be released on Canvas, initially for students who submit at the original due date, and perhaps later for students whose submission is delayed eg for simple extension.

The questions

A [relational algebra] (1 point)

Note: if you have difficulty producing relational algebra symbols, you may use text, with SEL_{C} to represent σ_C and PROJ_{D} to represent π_D and X JOIN Y to represent X ⋈ Y.

Consider the following relational schema which stores information about the selling operations of a store.

```
CREATE TABLE Part (PId INTEGER,
                   PDesc VARCHAR(30),
                   PUsualPrice INTEGER NOT NULL,
             PRIMARY KEY (PId));
CREATE TABLE Customer (CId INTEGER,
                       CName VARCHAR(20),
                       CAddr VARCHAR(40),
              PRIMARY KEY (CId));
CREATE TABLE Cust_Phone (CId INTEGER,
                         CPhone CHAR(15),
                         CPhoneFee INTEGER,
             PRIMARY KEY (CId, CPhone),
             FOREIGN KEY (CId) REFERENCES Customer(Cid));
CREATE TABLE Sale (SId INTEGER,
```

```
            SDate DATE,
            CId INTEGER NOT NULL,
            PId INTEGER NOT NULL,
            SPrice INTEGER NOT NULL,
      PRIMARY KEY (SId),
      FOREIGN KEY (CId) REFERENCES Customer(Cid),
      FOREIGN KEY (PId) REFERENCES Part(Pid));
```

In **Part** a row (31, 'Large Widget', 3050) represents that the part whose PId is 31, is described as "Large Widget" and the usual price is $30.50 (that is, PPrice is measured in cents). In **Customer**, a row (27, 'Jay Lui', '20 Pine St, Sydney') represents that the customer whose CId is 27 is named "Jay Lui", with address "20 Pine St, Sydney. In **Cust_Phone**, a row (27, '0414441777', 300) represents that the customer with CId 27 has a phone which can be reached as 0414662306 and the customer will pay a fee of $3 for each support call they make on that phone (CPhoneFee is measured in cents). In **Sale**, a row (110, '2022-01-20', 27, 31, 2990) represents that the sale whose SId is 110 occurred on 20 January 2022 when customer with CId 27 purchased the part with PId 31, for a price of $29.90 (SPrice is measured in cents).

Write *two different (but logically equivalent) relational algebra expressions* involving the tables in the schema above, so that each has a result which will answer the request ""Find the SId of every sale where the sale price is less than one half of the usual price of the part being sold"

B [relational design theory] (2 points)
Consider a schema which has one table the following relational design, which collects data about the air flights. There is a single table:
Crew(flightCode,destination,date,tailCode,acName,airline,emp_id,name,title).

In **Crew**, a tuple ('QF141', 'Dubai', '2019-10-31', 'VH-EAB', 'City of Canberra', 'Qantas', 65098, 'Joe Bloggs', 'Navigator') indicates that on 31st October 2019, flight QF141 departed for Dubai on Qantas aircraft VH-EAB ("City of Canberra") with navigator Joe Bloggs (employee ID 65098) on board. Note that tailCode is the government issued identifier for an aircraft. The following functional dependencies are valid in this schema:

- **flightCode → destination**
- **flightCode, date → tailCode**
- **tailCode → acName, airline**
- **emp_id → name, title**

B(i) Explain in English the meaning of the fd flightCode, date → tailCode. Also, give an example of data that would not be allowed in the table **Crew** because of this dependency.

B(ii) Based on this schema and its fds, calculate the attribute closure (flightcode, empid)+ Show the step-by-step working of the calculation.

B(iii) State whether the relation **Crew** is in BCNF. Justify your answer.

B(iv) *Give a lossless-join decomposition* of **Crew** into **two** relations. For each decomposed relation, *state the functional dependencies* that hold for it, and *state a primary key* for it. *Justify* that your decomposition has lossless-join property.

C. [index choices] (1 point)

Consider again the relational schema shown in Question A, and the Cust_Phone table. Suppose that no CREATE INDEX commands have been executed; that means that the structure of the table is as a tree-based clustering index on the primary key (CId, CPhone).

Consider the SQL query below, that refers to the sales schema shown on page 2.

```
SELECT CId
FROM Cust_Phone
WHERE CPhone = '0414441777' AND CPhoneFee > 500
```

C(i) Explain why this query cannot be answered effectively using the primary index, and state how the query would be answered with the initial structure of the table.

C(ii) Write SQL to create an index that will allow much faster execution for the query above, when the database has a large amount of data in the table. Also, explain how the index would allow the query to be calculated.

D. [Database-backed web-app security] (1 point)

In the code you used as a skeleton for asst3 (and which had previously been used in lab for week 8), access to the data in the database is given only to certain database accounts (the data owner, and any account to which the data owner granted permissions). In the following questions refer to the webapp as given in the skeleton, without any of your modifications.

D(i) Describe how the password for a database account is provided to the dbms, and what security protects this password.

D(ii) *Identify any weakness* in the security this provides, and *discuss* how this security mechanism could be improved.

Weekly progress meetings

Note that in week 11, the 20 minute meeting of your group with tutor is still about asst3. As assignment4 is individual work, the meeting of the group members with tutor in weeks 12 and 13 are used to allow students to raise any questions or seek clarification on any topics. To earn the contribution point from these meetings, a student should bring at least one question to ask the tutor.