CS2102: Database Systems

Assignment 02: SQL Queries

Deadline: Week 10

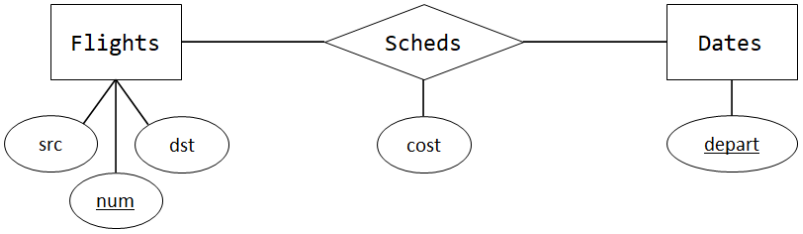
Instructions

- **Coursemology page:** `Assignments > Assignment 02`
- **Total marks:** 10 marks
- **Weightage:** 5%
- Submission on Coursemology page

1. Please read all the instructions below
2. All questions will appear on Coursemology in a single page
3. For questions involving drawing, you may submit a photograph instead of digital drawing but make sure that the photograph is clear
4. Submissions are not autograded as it will tell you the solution and can be brute-forced given enough time so instead, the grading will be done after deadline
5. Do not attempt the assignment at the last minute, since Coursemology may not be able to handle close to 400 simultaneous connections
6. Once submissions are finalized, it is considered final and we will not reopen the submission for you so be very careful when you finalize it
7. The deadline is firm, please finalize submission before the deadline
8. In all questions, you are restricted to at most 1 CTE
9. In all questions, you are not allowed to use the `VIEW` from previous questions/subquestions

Problem 1 : Airlines — 4 marks

Consider an airline with the following database tables and the corresponding ER diagrams:




```
CREATE TABLE flights (
  num   VARCHAR(10) PRIMARY KEY,
  src   VARCHAR(5) NOT NULL,
  dst   VARCHAR(5) NOT NULL
);
```

```
CREATE TABLE scheds (
  num     VARCHAR(10) REFERENCES flights(num),
  depart  DATE NOT NULL,
  cost    NUMERIC NOT NULL CHECK (cost > 0),
  PRIMARY KEY (num, depart)
);
```

We assume that an entry in `Scheds` means that the given flight number is scheduled to fly at the given date and with the given cost.

a) [1 marks] Find all flights numbers `(flight, connecting)` from Incheon (`'ICN'`) to Cengkareng (`'CGK'`). Include only flights with a single connecting flight. For instance, we include flights with the following itineraries: `ICN -> SIN -> CGK` or `ICN -> TPE -> CGK`. However, we should not include direct flights (*e.g.*, `ICN -> CGK`).

Write your answer as the following view:
`CREATE VIEW Q1 AS`

b) [3 marks] Find all flights schedules `(flight, flight_date, connecting, connecting_date, cost)` from Incheon (`'ICN'`) to Cengkareng (`'CGK'`). Include flights with a single connecting flight such that the connecting flight is at most 2 days after the first flight (and obviously the connecting flight cannot be before the first flight). Also include direct flights without any connecting flights (*e.g.*, include `ICN -> CGK`).

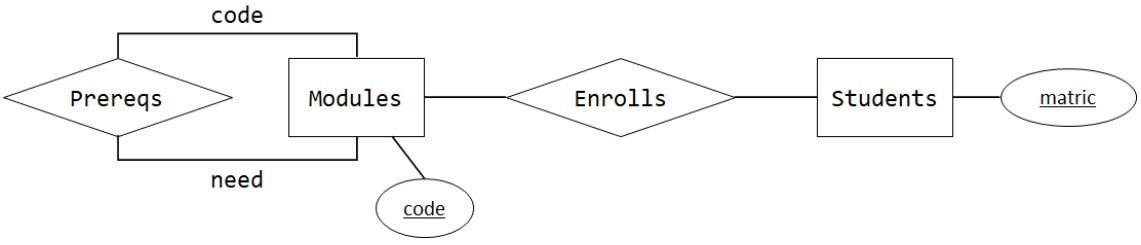
Write your answer as the following view:
`CREATE VIEW Q2 AS`

If there is no connecting flight (*e.g.*, direct flight), simply put `NULL` as the value for `connecting` and `connecting_date`. We let the `cost` to be the total cost (*e.g.*, for flight with connecting flight, it's the sum of the cost of both flights). Lastly, order the result in increasing value of `cost`.

HINT: To get the difference between two dates as the number of days, you can simply subtract them. Try the following SQL query: `SELECT date('2020-01-04') - date('2019-12-31');`

Problem 2 : Modules — 6 marks

Consider a university with the following database tables and the corresponding ER diagrams:




```
CREATE TABLE modules (
  code  VARCHAR(10) PRIMARY KEY
);
CREATE TABLE students (
  matric VARCHAR(9) PRIMARY KEY
);
CREATE TABLE prereqs (
  code  VARCHAR(10) REFERENCES modules(code),
  need  VARCHAR(10) REFERENCES modules(code),
  PRIMARY KEY (code, need)
```

```
);
CREATE TABLE enrolls (
  matric VARCHAR(9)  REFERENCES students(matric),
  code   VARCHAR(10) REFERENCES modules (code)
);
```

The roles for the `Prereqs` can be read as follows:

- The entry ⟨ `'CS1020'`, `'CS1010'` ⟩ means that the module with code `'CS1020'` has the module with code `'CS1010'` as its pre-requisites.
- The entry ⟨ `'CS2102'`, `'CS1020'` ⟩ and ⟨ `'CS2102'`, `'CS1231'` ⟩ (*on different rows*) means that the module with code `'CS2102'` has the modules with code `'CS1020'` and `'CS1231'` as its pre-requisites.

We assume that an entry in `Enrolls` means that the student has passed the given module.

a) [2 marks] Find all module codes (`code`) that can be taken by the student with matric number `'A0000001A'`. In other words, all the pre-requisites have been fulfilled and the module has not been taken before.

Write your answer as the following view:
`CREATE VIEW Q3 AS`

b) [2 marks] Find all students (`matric`) that can take the module `'CS2102'`. In other words, all the pre-requisites have been fulfilled and the module has not been taken before.

Write your answer as the following view:
`CREATE VIEW Q4 AS`

c) [2 marks] We define the concept of *level* of the module. We let modules without any pre-requisites to be module at level 0. If a module M1 requires another module M2 such that M2 is at level n, then module M1 is a module at level n+1. We assume that level n is the largest level of the pre-requisites of module M1.

For instance, `'CS1010'` is a module at level 0. Hence, `'CS1020'` is a module at level 1. As for `'CS2102'`, it has two pre-requisites: `'CS1231'` which is a level 0 module and `'CS1020'` which is a level 1 module. Hence, `'CS2102'` is a module at level 2.

Assume that the largest level is a module at level 4. Find all distinct module codes (`code`) that a student need to pass before they can take the module `'CS4221'`.

Write your answer as the following view:
`CREATE VIEW Q5 AS`