UIT2201: CS & the IT Revolution
Tutorial Set 7 (Fall 2016)

(D-Problems discussed on Friday, 30-Sep-2016)
(Q-Problems due on Tuesday, 04-Oct-2016)

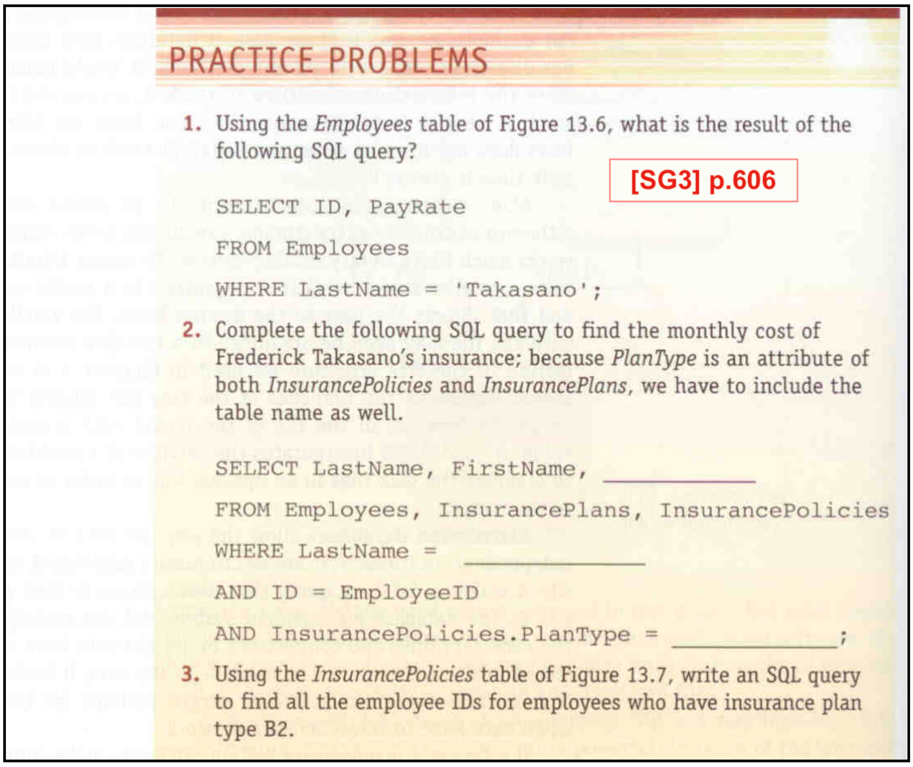
Note: Fig 13.6 and 13.7 of [SG3] are tables `EMPLOYEES`, `INSURANCEPOLICIES`, and `INSURANCEPLANS` for relational database model of Rugs-For-You. They can be found in the lecture notes.

Practice Problems: (not graded)

These are practice problems for you to try out. *(If you have difficulties with these practice problems, please **quickly** see your classmates or the instructor for help.)*

T7-PP1: (SQL Query) Read Chapter 13.3 (pp 598-606) of [SG3].

T7-PP2: (SQL Query) Problems 1, 2, 3 on page 606 (Chapter 13) of [SG3].



PRACTICE PROBLEMS

[SG3] p.606

1. Using the *Employees* table of Figure 13.6, what is the result of the following SQL query?

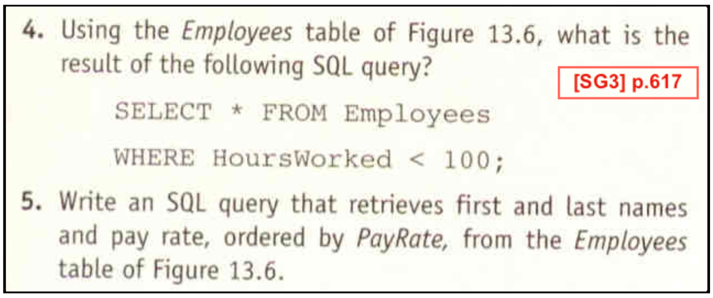
```
SELECT ID, PayRate
FROM Employees
WHERE LastName = 'Takasano';
```

2. Complete the following SQL query to find the monthly cost of Frederick Takasano's insurance; because *PlanType* is an attribute of both *InsurancePolicies* and *InsurancePlans*, we have to include the table name as well.

```
SELECT LastName, FirstName, ____________
FROM Employees, InsurancePlans, InsurancePolicies
WHERE LastName = ____________
AND ID = EmployeeID
AND InsurancePolicies.PlanType = ____________;
```

3. Using the *InsurancePolicies* table of Figure 13.7, write an SQL query to find all the employee IDs for employees who have insurance plan type B2.

T7-PP3: (SQL Query) Problems 4, 5 on page 617 (Chapter 13) of [SG3].



4. Using the *Employees* table of Figure 13.6, what is the result of the following SQL query?

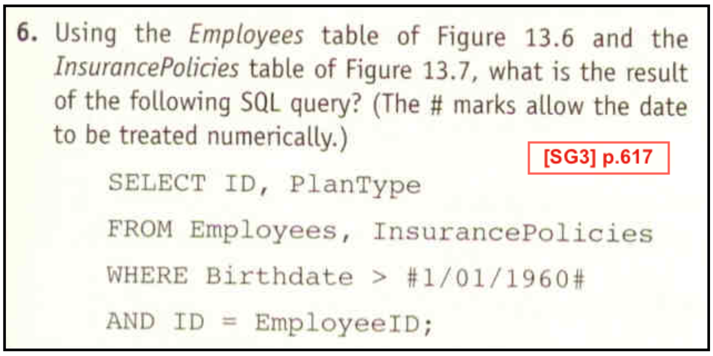
[SG3] p.617

```
SELECT * FROM Employees
WHERE HoursWorked < 100;
```

5. Write an SQL query that retrieves first and last names and pay rate, ordered by *PayRate*, from the *Employees* table of Figure 13.6.

Discussion Problems: -- Prepare (individually) for tutorial discussion.

T7-D1: (SQL Query)
Problems 6 on page 617 (Ch. 13) of [SG].
(**Note:** First read Ch.13.3 of [SG3] to learn about SQL.)



6. Using the *Employees* table of Figure 13.6 and the *InsurancePolicies* table of Figure 13.7, what is the result of the following SQL query? (The # marks allow the date to be treated numerically.)

[SG3] p.617

```
SELECT ID, PlanType
FROM Employees, InsurancePolicies
WHERE Birthdate > #1/01/1960#
AND ID = EmployeeID;
```

T7-D2: (Efficient Query Processing) -- given in next page.

T7-D3: (Question from a former Quiz) --
Question Q4 from [Spring-2013 MidTerm-Q4].
(Ignore Q4(a)(ii) and Q4(b)(ii) on use of basic database primitives.)

Problems to be Handed in for Grading by the Deadline:

(**Note:** Please submit *hard copy* to me. Not just soft copy via email.)

T7-Q1: (10 points) (SQL Query) [Modified from Problems 7, p625, Ch 13 of [SG].]
(a) Using the `Employees` table of Figure 13.6 and the `InsurancePolicies` table of Figure 13.7, write an SQL query that retrieves `ID, FirstName, LastName, PlanType, DataIssued` for all employees have pay rate of more than $10.00.

(b) Suppose we use the `JOIN` operation on `Employees` and `InsurancePolicies` to process this SQL query; similar to that given in the lecture notes (2016-04a-SQL-anim.ppt, Query Q3).
How many *row operations* are needed to process this SQL query?

(c) Show the output table produced by this SQL query. How many rows are there in this table?

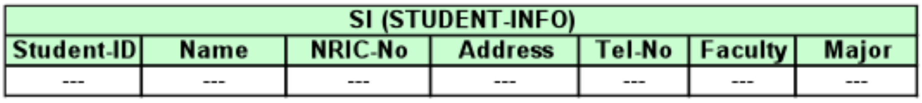
T7-Q2: (20 points) (Efficient Query Processing) -- given in next page.

A-Problems: OPTIONAL Challenging Problems for Further Exploration

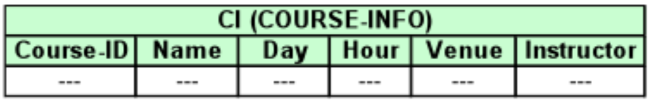
A6-2016: (Really LARGE numbers -- how to deal with them!)
Some entries in the table for T5-Q4 cause overflow in Excel/calculators -- and are listed as "too big to compute". (Example, for $T(n)=2^n$, when n=1000, the running time is 3.40 x 10^{291} yrs.)
Explore *creative* way to compute these *very big numbers with the help of calculators*.
(*Hint:* John Napier, 1614)

Consider a database with 3 tables that will be used for the problems T7-D2, D3, Q2, and Q3. `STUDENT-INFO, COURSE-INFO, and ENROLMENT.` We shall assume that

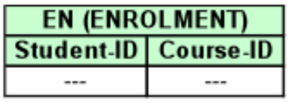
- the `SI=STUDENT-INFO` table has 30,000 (3x10^4) rows,
- the `CI=COURSE-INFO` table has 1,000 (10^3) rows, [BiYing checked CORS & said 1365 for Spr 2009. Thx]
- the `EN=ENROL=ENROLMENT` table has 100,000 (10^5) rows.



SI (STUDENT-INFO)						
Student-ID	Name	NRIC-No	Address	Tel-No	Faculty	Major
---	---	---	---	---	---	---



CI (COURSE-INFO)					
Course-ID	Name	Day	Hour	Venue	Instructor
---	---	---	---	---	---



EN (ENROLMENT)	
Student-ID	Course-ID
---	---

T7-D2: (Efficient Query Processing) [Note: *First* read notes and also do T7-D1.]
(a) Give a "concise English description" of the output of the following SQL query:

```
SELECT Name, Faculty, Course-ID
FROM SI, EN
WHERE (Student-ID='U2908888P') and (SI.Student-ID = EN.Student-ID)
```

(b) Give an "SQL query" statement to obtain each of the following:

i. List the **`Student-ID, Name, Tel-No`** of the all `"CS"` majors;
ii. List the **`Student-ID, Name, Tel-No`** of the students enrolled in course `"UIT2201"`;
iii. List the **`Student-ID, Name, Tel-No`** of students taught by instructor `"LeongHW"`.

(c) How many *row operations* are needed to process each of the SQL queries given above?
(Note: You will need to estimate of the size of the intermediate table for (iii).)

T7-Q2: (20 points) (Continued from T7-D2 above)

(a) Give a "concise English description" of the output of the following SQL query:

```
SELECT Course-ID, Name, Instructor
FROM CI, EN
WHERE (Student-ID='U2908888P') and (CI.Course-ID = EN.Course-ID)
```

(b) Give an "SQL query" statement to obtain each of the following:

i. List the **`Student-ID, Major, Course-ID`** of all courses taken by `History` majors;
ii. List the **`Student-ID, SI.Name, Tel-No`** of all `History` majors;
iii. List the **`Student-ID, SI.Name, Tel-No`** of all `History` majors who have lectures in `"LT13"`.

(c) How many *row operations* are needed to process each of the SQL queries given above?
(Note: You will need to estimate of the size of the intermediate table for (iii).)

UIT2201: CS & IT Revolution; (Fall 2016); A/P Leong HW