Jacob M. Peck
CSC 459 - Databases
Professor James P. Early
February 10, 2012

Assignment 2

The following schema will be used for all questions.
$employee = (emp\_code, name, job\_code, effective\_date)$
$benefit = (emp\_code, plan\_code)$
$job = (job\_code, job\_description, exempt)$
$plan = (plan\_code, plan\_description)$

Notes: employees can have multiple jobs over time. The attribute "exempt" has a Y/N value (meaning, they are exempt from overtime pay).
Write relational algebra queries for each the following.

1. **Q** Show the employee code, name, and effective date for all employees.

   A $\Pi_{emp\_code,name,effective\_date}(employee)$

2. **Q** Show the names and job description for all employees in a job classified as exempt.

   A $\Pi_{name,job\_description}(employee \bowtie \sigma_{exempt=Y}(job))$

3. **Q** Show the names, employee codes, and job codes of all employees who have held both exempt and non-exempt jobs over time.

   A $t1 \leftarrow employee \bowtie \sigma_{exempt=Y}(job)$
   $t2 \leftarrow employee \bowtie \sigma_{exempt=N}(job)$
   $\Pi_{name,emp\_code,job\_code}(\sigma_{t1.emp\_code=t2.emp\_code}(t1 \times t2))$

4. **Q** Show each job description, plan code, and description of benefit plans associated with employees in that job.

   A $t1 \leftarrow \Pi_{job\_code,job\_description}(job)$
   $t2 \leftarrow \Pi_{emp\_code,job\_code}(employee)$
   $t3 \leftarrow ((t1 \bowtie t2) \bowtie benefit) \bowtie plan$
   $\Pi_{job\_description,plan\_code,plan\_description}(t3)$

5. **Q** Show the employee name and effective date of all employees in their **current** job. Note: you may assume that when comparing dates, the later date is greater than earlier date.

   A $t1 \leftarrow \Pi_{name,effective\_date}(employee)$
   $t2 \leftarrow \rho_{t2}(t1)$
   $t3 \leftarrow \Pi_{name,effective\_date}(\sigma_{name=t2.name \wedge effective\_date<t2.effective\_date}(t1 \times t2))$
   $t1 - t3$

6. **Q** Suggest a query that would be greatly simplified by using a division operation, and show the relational algebra syntax.

   A One such query could be "Find the names of all employees who have held 3 jobs: those with job_code 134, 167, and 184."

   $t1 \leftarrow \rho_{t1(job\_code)}(\{(134), (167), (184)\})$
   $\Pi_{name}(employee \div t1)$

7. **Q** Show how you can use an outer join to identify the names of all employees who do not have a benefit plan.

   A $\Pi_{name}(\sigma_{plan\_code=null}(employee \rtimes benefit))$

8. **Q** Given these queries, suggest and explain some candidate keys that could be used for each relation.

   A For the employee relation, a good candidate key would be the tuple consisting of $(emp\_code, job\_code, effective\_date)$, as this would ensure that every job that every row could be identified, regardless of how many jobs the employee had at one time or how many times they had the same job, even.

   For the benefit relation, you could use the tuple $(emp\_code, plan\_code)$, as a single employee might have more than one plan, and each plan will most likely be associated with multiple employees.

   For the job relation, I'd suggest using the tuple $(job\_code, exempt)$, as some employees may be tax exempt regardless of their job.

   For the plan relation, I'd use the $plan\_code$ attribute as a key. There is very little chance that the same plan code would be used for two different plans, so there is no risk here.