Overview: This assignment focuses on a study of *windowing queries* in general, and the query from the midterm exam and implementations in PostgreSQL in particular. In addition to expressing the query in SQL, we also study its algebraic forms using logical and physical operators, and some performance aspects in PostgreSQL based on an experimental study. This assignment is open ended and we will use the newsgroup for questions and discussions on the details. An important part of this assignment is clearly documenting and presenting the results, so well documented code is especially important.

Recall the battery-charging database:

Batteries

tag varchar(50)	model varchar(50)	buy_date date	price float	manuf_date date	color varchar(50)	notes varchar(50)
x1	Tenergy AB	2009-01-23	2.20	2008-06-01	blue	heavy
x2	Tenergy AB	2009-01-23	2.20	2008-06-01	blue	light
pq	Tenergy AB	2010-10-03	2.25	2009-06-01	blue	check
pq2	Tenergy AB	2010-10-30	2.50	2009-06-01	blue	

Charges

tag varchar(50)	model varchar(50)	charger varchar(50)	date date	mAh float	method varchar(50)
x1	Tenergy AB	maha-101	2012-01-22	1883	charge
x1	Tenergy AB	maha-101	2012-02-12	1983	refresh

Models

m_id varchar(50)	manuf varchar(50)	model varchar(50)	rated_mAh float	notes varchar(50)
Tenergy AB	Tenergy	Essential	2500	
Amazon B	Amazon	Basics	2200	OEM unknown

For notational convenience in relational algebra, we may abbreviate as follows:

`Batteries(tag, model, buy_date, price, manuf_date, color, notes)`	$B(T, M, B, P, F, C, N)$
`Charges(tag, model, charger, date, mAh, method)`	$C(T, M, C, D, A, E)$
`Models(m_id, manuf, model, rated_mAh, notes)`	$M(I, F, M, A, N)$

For SQL-based questions, provide standard SQL answers. If necessary, modify your answers to work with PostgreSQL and clearly note the changes that were necessary. Submit all answers in electronic form.

1. Provide SQL statements that create the above database, including schema and instance.

2. Briefly describe an efficient method for loading a large amount of data into the above database. Use this method to create a script that loads a suitably large and representative dataset that will be used for testing and experiments. Include this script

and supporting material in your submission. Describe and justify your method for generating the test dataset.

3. Express the following query (from the midterm exam) in SQL, using windowing functions: batteries with the maximum reduction in measured capacity within a calendar month.

4. Recast the query of Question 3 without using windowing functions and related features.

5. Express the queries of Questions 3 and 4 using extended bag algebra. For the query of Question 3, you may augment the algebra with a suitable operator. Clearly define any new operators you use and justify their definitions.

6. Determine the query plans used by PostgreSQL for the queries of Questions 3 and 4. Depict the query plans neatly using a conventional tree representation.

7. Conduct experiments to determine the impact of indexes on the query plans chosen by PostgreSQL for your sample database instance. Determine the collection of indexes that yields the best query plan for the queries of Questions 3 and 4. Document your experimental procedure and results carefully. Wherever possible, include scripts that may be used to re-run your experiments (e.g., scripts that create and drop various indexes in turn, to study the query plans). Make appropriate use of tables and charts.

Submission: Please follow the submission guidelines from the previous assignment, replacing `hw02` with `hw03` where appropriate. Be certain to check file-naming conventions, required README and Makefile contents. Use the newsgroup to for clarification on question details, submission formats, and other requirements.