Preparation for the Final

Notes

This study guide is built from a collection of CSC 365 labs. In these labs in CSC 365, the students were asked to write SQL queries to retrieve the necessary information from the relational databases storing the data from a variety of provided datasets.

For CSC 369, I have chosen a few of the datasets with only 2-4 tables in each. I have provided verbatim the information needs from a series of CSC 365 labs. Your goal is to render these information needs as MapReduce pseudocode with input files in the CSV format.

Since you are working with pseudocode you can assume straightforward parsing of the input (use the "dot" notation to refer to the attributes of each input record in Map() for example).

You may need to set up more than one MapReduce to correctly implement some of the information needs. Watch for the join patterns.

The actual assignments on the final exam will be of moderate complexity, so some of the most complicated information needs in this study guide will go way beyond what is expected on the final exam. Please remember that.

There is no expectation that you solve all of these problems before coming to the final exam. Rather, the sheer number of problems is provided to give you an opportunity to pick and choose the ones you want to try your hand at solving. There is going to be some repetition of solutions, which is good for you as well, as it makes the MapReduce design patterns more pronounced and memorable.

Datasets

Of the nine datasets used for CSC 365, I have selected three for you: BAKERY, INN and WINE. This is done primarily because of the number of tables

in these datasets. The dataset descriptions are below. The course web page contains the links to the CSV files and the README documentation, where all attributes of all CSV files are specified.

BAKERY dataset. Type: OLTP (on-line transaction processing) dataset. This dataset records information about one month of sales from a small bakery to a list of its dedicated customers. The dataset captures the notions of a transaction (a single purchase) and market baskets (each purchase may contain more than one item). The dataset contains four files:

1. `customers.csv` - list of 20 customers of the bakery.
2. `goods.csv` - list of 50 items on the bakery's menu.
3. `receipts.csv` - list of 1000 purchases made at the bakery during one month period.
4. `items.csv` - list of actual goods purchases (for each receipt) from the bakery.

WINE dataset. Type: normalized (somewhat). The dataset lists ratings of a variety of single-grape California wines of different vintages as given by the Wine Spectator magazine. The dataset consists of a list of wines complete with their ratings on a 100 point scale, their reported sales prices and the production volumes. Two additional lists: appelation/American Viticultural Areas (AVAs) and grape varieties are available as well. The dataset contains three files:

1. `appellations.csv` - list of appelations (wine producing areas) for California wines.
2. `grapes.csv` - list of grapes used in winemaking.
3. `wine.csv` - list of wines with their characteristics and production information.

INN dataset. Type: simple OLTP. The dataset contains information about one year's (2010) worth of completed hotel reservations for a small Bed & Breakfast inn. The inn contains 10 uniquely named rooms - each with its own set of features. The dataset lists all reservations that *commenced* 2010[1] and specifies for each room reservation the checkin and checkout dates, the name of the person making the reservation and the number of adults and kids staying in the room. The dataset consists of two data files:

1. `Rooms.csv` - list of the 10 rooms in the Bed&Breakfast.
2. `Reservations.csv` - list of all reservations at the Bed&Breakfast in 2010.

[1] Some reservations made at the end of the year end in 2011.

Information Needs

For each information need construct pseudocode for a its implementation via one or more MapReduce processes. Where necessary indicate the order in which Map() and Reduce() processes need to run, and specify the appropriate file inputs to all Map() functions.

BAKERY dataset

1. Find all chocolate-flavored items on the menu whose price is under $5.00. For each item output the flavor, the name (food type) of the item, and the price. Sort your output in descending order by price.

2. Report the prices of the following items:

   - any cookie priced above $1.10;
   - any lemon-flavored items;
   - any apple-flavored item except for the pie.

   Output the flavor, the name (food type) and the price of each pastry. Sort the output in alphabetical order by the flavor and then the name.

3. Find all customers who made a purchase on October 3, 2007. Report the name of the customer (first, last). Sort the output in alphabetical order by the customer's last name. Each customer name must appear at most once.

4. Find all different cakes purchased on October 4, 2007. Each cake (flavor, food) is to be listed once. Sort output in alphabetical order by the cake flavor.

5. List all pastries purchased by `ARIANE CRUZEN` on October 25, 2007. For each pastry, specify its flavor and type, as well as the price. Output the pastries in the order in which they appear on the receipt (each pastry needs to appear the number of times it was purchased).

6. Find all types of cookies purchased by `KIP ARNN` during the month of October of 2007. Report each cookie type (flavor, food type) exactly once in alphabetical order by flavor.

7. Find all dates in the first half of October of 2007 (October 1 to October 15 inclusive) on which one customer made multiple purchases. Report each date exactly once, output dates sorted in ascending order.

8. Find all customers who purchased, during the same trip to the bakery, two different `Croissant`s. Report first and last names of the customers in alphabetical order by their last name.

9. Find all customers who did not make a purchase between October 14 and October 19, 2007 (inclusively). Report their first and last names sorted alphabetically by last name.

10. Find all days on which *either* `ALMETA DOMKOWSKI` made a purchase, *or* someone purchased a `Gongolais Cookie`. Sort dates in chronological order. Each date shall appear exactly once.

11. Report the total amount of money `NATACHA STENZ` spent at the bakery during the month of October, 2007.

12. Report the total amount of money spent by bakery customers in October 2007 on `Cookie`s.

13. For each pastry flavor which is found in more than three types of pastries sold by the bakery, report the average price of an item of this flavor and the total number of different pastrties of this flavor on the menu. Sort the output in ascending order by the average price.

14. Find the total amount of money the bakery earned in October 2007 from selling eclairs. Report just the amount.

15. For each purchase made by `NATACHA STENZ` output the receipt number, the date of purchase, the total number of items purchased and the amount paid. Sort in descending order by the amount paid[2].

16. For each day of the week of October 8 (Monday to Sunday) report the total number of purchases (receipts), the total number of pastries purchased and the overall daily revenue. Report results in chronological order and include both the day of the week and the date. [3].

17. Report all days on which more than ten tarts were purchased, sorted in chronological order.

18. Find the customer(s) who spent the most on pastries in October of 2007. Report first and last name.

19. Find the customers who never purchased an eclair (`'Eclair'`) (in October of 2007). Report their first and last names in alphabetical order by last name.

20. Find the type of baked good (food type, flavor) responsible for highest total revenue.

21. Find the most popular (by number of pastries sold) item. Report the item (food, flavor).

22. Find the day of the highest revenue in the month of October, 2007.

23. For every customer who DID NOT make a purchase on the day of the highest revenue, report the total number of purchases (overall) the customer made and the last date of a purchase. Order the output by the total amount of purchases.

[2]The total amounts paid may look strange, if you are using floating points for prices.
[3]The total amounts paid may look strange, if you are using floating points for prices.

24. for every type of `Croissant` report the customer(s) who purchased it the largest number of times during the month of October 2007. Report the name of the pastry (flavor, food type), the name of the customer (first, last), and the number of purchases made. Sort output in descending order on the number of purchases, then in alphabetical order by last name of the customer.

25. Output the names of all customers who did not make a purchase between October 20 and October 24 (inclusive) of 2007. Output first and last names in alphabetical order by last name.

26. Output the names of all customers who made multiple purchases (more than one receipt) on the latest day in October on which they made a purchase. Report names (first, last) of the customers and the earliest day in October on which they made a purchase, sorted in chronological order.

27. Find which types of cakes were never purchased on Mondays. Report full description (flavor, food) in alphabetical order.

INN dataset

1. Find all `modern` rooms with a base price below $160 and two beds. Report room names and codes in alphabetical order by the code.

2. Find all July reservations (a.k.a., all reservations that both start AND end in August) for the 'Convoke and sanguine' room. For each reservation report the last name of the person who reserved it, checkin and checkout dates, the total number of people staying and the daily rate. Output reservations in chronological order.

3. Find all rooms occupied on February 6, 2010. Report full name of the room, the check-in and checkout dates of the reservation. Sort output in alphabetical order by room name.

4. For each stay of `GRANT KNERIEN` in the hotel, calculate the total amount of money, he paid. Report reservation code, checkin and checkout dates, room name (full) and the total amount of money the stay cost. Sort output in chronological order by the day of arrival.

5. For each reservation that starts on December 31, 2010 report the room name, nightly rate, number of nights spent and the total amount of money paid. Sort output in descending order by the number of nights stayed.

6. Report all reservations in rooms with double beds that contained four adults. For each reservation report its code, the full name and the code of the room, check-in and check out dates. Report reservations in chronological order, and sorted by the three-letter room code (in alphabetical order) for any reservations that commenced on the same day.

7. Find all rooms that were occupied on all three of the following dates: May 15, 2010, August 18, 2010 and December 12, 2010. Report just the full name of the room and the room code. Sort output in alpabetical order by room name.

8. Find the names of all people[4] staying at the inn at the same time as `HERBERT FRYDAY`. Sort the output in alphabetical order by last name.

9. Find the number of August reservations (both checkin and checkout dates are in August) where two adults are staying with two children.

10. Find the average number of nights of stay in the `'Interim but salutary'` room for all reservations that commenced May 1, 2010 or later and ended before August 31, 2010.

11. For each room report the total revenue for all stays and the average revenue per stay generated by stays in the room that originated in the months of September, October and November. Sort output in descending order by total revenue. (Output full room names).

12. Report the total number of reservations that commenced on Fridays and the total revenue they brought in. (*Hint:* look up the date of the *first* Friday on the calendar).

13. For each day of the week, report the total number of reservations commenced on it and the total revenue these reservations brought. Report days of week as `Monday`, `Tuesday`, etc.

14. For each room report the highest markup against the base price and the smallest markup (i.e., largest markdown). Report markups and markdowns in absolute terms (absolute difference between the base price and the rate). Sort output in descending order by the absolute value of the largest markup. Report full names of the rooms.

15. For each room report how many nights in 2010 the room was occupied. Report the room code, the full name of the room and the number of occupied nights. Sort in descending order by occupied nights. (Note: it has to be *number of nights in 2010* - the last reservation in each room *may* and *will* can go beyond December 31, 2010, so the "extra" nights in 2011 need to be deducted).

    Note/Hint: This is almost an extra credit problem. While multiple solutions are possible, my solution uses SQL's `SIGN()` built-in function which returns -1 for negative numbers, +1 for positive numbers and 0 for 0.

16. Find the least popular room in the hotel. The least popular room is the room that had seen the lowest number of reservations (Note: if there is a tie for the least popular room status, report all the least

[4] We only know the names of the people who made the reservations, so only those names are subject to the query.

popular rooms). Report the full name of the room, the room code and the number of reservations.

17. Find the room that has been occupied the most based on the reservations in the database[5]. Report the room name, room code and the number of days it was occupied.

18. Find the most expensive reservation(s) made. Report the room name (full), dates of stay, last name of the person who made the reservation, daily rate and the total amount paid.

19. For each room, report the most expensive reservation. Report the full room name, dates of stay, last name of the person who made the reservation, daily rate and the total amount paid. Sort the output in descending order by total amount paid.

20. Find the best month (i.e., month with the highest total revenue). Report the month, the total number of reservations and the revenue. For the purposes of the query, count the entire revenue of a stay that commenced in one month and ended in another towards the earlier month. (e.g., a September 29 - October 3 stay is counted as September stay for the purpose of revenue computation).

21. For each room report whether it is occupied or unoccupied on October 22, 2010. Report the full name of the room, the room code, and put either `'Occupied'` or `'Empty'` depending on whether the room is occupied on that day. (the room is occupied if there is someone staying the night of May 19, 2010. It is NOT occupied if there is a checkout on this day, but no checkin). Output in alphabetical order by room code.

22. For each room report how many reservations were made for the most expensive rate for that room. Report full room name and the appropriate number of reservations. Sort the output in ascending order by the number of reservations.

WINE dataset

1. List all AVAs located in Monterey county. Output just the names of the AVA appellations and sort them in alphabetical order.

2. List all white grape varieties for which at least one wine of the 2008 vintage is rated at 90 points or above in the database. Each grape variety needs to be reported once. Sort the output in alphabetical order.

3. List all Sonoma county appellations for which the database contains at least one rating for a `'Grenache'`. For each appellation list its name and county. Sort output in alphabetical order by county, then by appellation name. Report each appellation once.

[5] No need to limit the number of occupied days to 2010.

4. List all vintage years in which at least one `Zinfandel` from Sonoma County (any appellation) scored above 92. Each year needs to be reported once. Sort in chronological order.

5. A case of wine is 12 bottles. For each `Carlisle` (name of the winery) `Syrah` compute the total revenue assuming that all the wine sold at the specified price. Report the name of the wine, its vintage wine score and overall revenue. Sort in descending order by revenue. Exclude NULL values.

6. Compute the total price of a bottle of `Kosta Browne`'s `Koplen Vineyard` 2008 `Pinot Noir`, two bottles of `Darioush`'s 2007 `Darius II Cabernet Sauvingnon` and a bottle of `Kistler`'s `McCrea Vineyard` 2006 `Chardonnay`. Report just the one number.

7. List all 2006 vintage wines from Napa (any appellation within the county) whose total revenue exceeds that of the 2006 `'Appelation Series'`[6] Paso Robles Zinfandel from `'Rosenblum'` winery. For each wine report grape, winery and name, score and revenue. Order by revenue.

8. Find all wines produced in the same vintage year as the `Tor Chardonnay`, which have both the higher score and the higher production.

9. Find the average score of a `Paso Robles Zinfandel`.

10. Find the total revenue from all red wines made by `Kosta Browne`.

11. Find the average number of cases of a `Pinor Noir` produced from grapes sourced from the `Central Coast`.

12. Report the overall number of different red wines produced in `Russian River Valley` during the year when this AVA had a wine with a score of 98.

13. For each wine score value above 88, report average price, the cheapest price and the most expensive price for a bottle of wine with that score (for all vintage years combined), the total number of wines with that score and the total number of cases produced. Sort by the wine score.

14. For each year, report the total number of red Sonoma County wines whose scores are 90 or above. Output in chronological order.

15. For each appellation that produced more than two Cabernet Sauvingnon wines in 2007 report its name and county, the total number of Cabernet Sauvingnon wines produced in 2008, the average price of a bottle of Cabernet Sauvingnon from that vintage, and the total (known) number of bottles produced[7]. Sort output in descending order by the number of wines.

[6] There is a typo there. Let it be for now.
[7] Recall, one case is 12 bottles.

16. For each appellation inside Central Coast compute the total (known)[8] sales volume that it can generate for the wines produced in 2008. Sort the output in descending order by the total sales volume. (Note: recall what a case of wine is).

17. For each county in the database, report the score of the highest ranked 2009 red wine. Exclude wines that do not have a county of origin (`'N/A'`). Sort output in descending order by the best score.

18. Find the grape(s) that grow(s) in the largest number of appellations. Report grape name, color and the number of appellations it grows in.

19. Find the most popular white grape (i.e., the grape that is used to make the largest number of white wines in the database). Report the name of the grape.

20. Report the grape with the largest number of high-ranked wines (score of 93 and above).

21. Report the appellation responsible for the largest number of high-ranked wines (score of 93 and above). Report just the name of the appellation.

22. Find the high-ranked wine (score of 93 or above) responsible for highest sales revenue. Report the vintage year, winery, wine name, score and the computed revenue.

23. Find if there are any 2008 Chardonnays that scored better than any 2007 Syrah. report winery, wine name, appellation, score and price.

24. Two California AVAs, Carneros and Dry Creek Valley have a bragging rights contest every year: the AVA that produces the highest-ranked wine among all the wines produced in both AVAs wins. Based on the data in the database, output (as a single tuple) the number of vintage years each AVA has won between 2005 and 2009 (you want the output to look like a score of a game between the two AVAs. Only the vintage years where one AVA won count - vintages when both AVAs had the same highest score should not be counted).

25. Find how many cases were produced of the most expensive red wine from San Luis Obispo county.

Submission Instructions

You must submit all your files in a single archive. Accepted formats are `gzipped tar` (`.tar.gz`) or `zip` (`.zip`). The file you are submitting must be named `lab4.ext` where `ext` is one of the extensions above. The archive shall contain eight directories: `AIRLINES`, `CARS`, `CSU`, `INN BAKERY`, `STUDENTS`, `MARATHON` and `WINE`.

Each directory shall contain the following SQL scripts:

[8] Recall, that information about production volumes for some wines is not available.

- Database creation script. (e.g., `CARS-setup.sql`). Use the scripts from Lab 2/Lab 4 submissions.
- Table creation script. Use `<DATASET>-insert.sql` (e.g., `CARS-insert.sql`) file from Lab 4 submission.
- The cleanup script (e.g., `CARS-cleanup.sql`). Use the scripts from Lab 2/Lab 4.
- **NEW script.** One script per database, containing all SQL statements and any SQL*plus statements needed for formatting. Name the script (as specified above) `<DATASET>-info.sql` (e.g., `CARS-info.sql`.

Submit using handin:

Section 01:

```
$ handin dekhtyar lab05-01 <file>
```

Section 03:

```
$ handin dekhtyar lab05-03 <file>
```