CHAPTER 22

Advanced Querying and Information Retrieval

Exercises

22.2 Answer:

```
groupby rollup(a), rollup(b), rollup(c), rollup(d)
```

22.4 Answer: We assume that multiple students do not have the same marks since otherwise the question is not deterministic; the query below deterministically returns all students with the same marks as the n student, so it may return more than n students.

```
select student, sum(marks) as total,
          rank() over (order by (total) desc) as trank
from S
groupby student
having trank ≤ n
```

22.6 Answer:

```
select t1.account-number, t1.date-time, sum(t2.value)
from transaction as t1, transaction as t2
where t1.account-number = t2.account-number and
          t2.date-time < t1.date-time
groupby t1.account-number, t1.date-time
order by t1.account-number, t1.date-time
```

22.8 Answer:

```
(select color, size, sum(number)
 from sales
 groupby color, size
)
union
(select color, 'all', sum(number)
 from sales
 groupby color
)
union
(select 'all', size, sum(number)
 from sales
 groupby size
)
union
(select 'all', 'all', sum(number)
 from sales
)
```

22.10 Answer: Consider the following pair of rules and their confidence levels :

No.	**Rule**	**Conf.**
1.	$\forall\ persons\ \ P,\ 10000 < P.salary \leq 20000 \Rightarrow P.credit = good$	60%
2.	$\forall\ persons\ \ P,\ 20000 < P.salary \leq 30000 \Rightarrow P.credit = good$	90%

The new rule has to be assigned a confidence-level which is between the confidence-levels for rules 1 and 2. Replacing the original rules by the new rule will result in a loss of confidence-level information for classifying persons, since we cannot distinguish the confidence levels of perople earning between 10000 and 20000 from those of people earning between 20000 and 30000. Therefore we can combine the two rules without loss of information only if their confidences are the same.

22.13 Answer: In a destination-driven architecture for gathering data, data transfers from the data sources to the data-warehouse are based on demand from the warehouse, whereas in a source-driven architecture, the transfers are initiated by each source.

The benefits of a source-driven architecture are

- Data can be propagated to the destination as soon as it becomes available. For a destination-driven architecture to collect data as soon as it is available, the warehouse would have to probe the sources frequently, leading to a high overhead.
- The source does not have to keep historical information. As soon as data is updated, the source can send an update message to the destination and forget the history of the updates. In contrast, in a destination-driven archi-

tecture, each source has to maintain a history of data which have not yet been collected by the data warehouse. Thus storage requirements at the source are lower for a source-driven architecture.

On the other hand, a destination-driven architecture has the following advantages.

- In a source-driven architecture, the source has to be active and must handle error conditions such as not being able to contact the warehouse for some time. It is easier to implement passive sources, and a single active warehouse. In a destination-driven architecure, each source is required to provide only a basic functionality of executing queries.
- The warehouse has more control on when to carry out data gathering activities, and when to process user queries; it is not a good idea to perform both simultaneously, since they may conflict on locks.

22.14 Answer:

```
select store-id, city, state, country,
        date, month, quarter, year,
        sum(number), sum(price)
from sales, store, date
where sales.store-id = store.store-id and
        sales.date = date.date
groupby rollup(country, state, city, store-id),
        rollup(year, quarter, month, date)
```

22.15 Answer: We do not consider the questions containing neither of the keywords as their relevance to the keywords is zero. The number of words in a question include stop words. We use the equations given in Section 22.5.1.1 to compute relevance; the log term in the equation is assumed to be to the base 2.

Q#	#wo- -rds	# "SQL"	#"rela- -tion"	"SQL" term freq.	"relation" term freq.	"SQL" relv.	"relation" relv.	Total relv.
1	84	1	1	0.0170	0.0170	0.0002	0.0002	0.0004
4	22	0	1	0.0000	0.0641	0.0000	0.0029	0.0029
5	46	1	1	0.0310	0.0310	0.0006	0.0006	0.0013
6	22	1	0	0.0641	0.0000	0.0029	0.0000	0.0029
7	33	1	1	0.0430	0.0430	0.0013	0.0013	0.0026
8	32	1	3	0.0443	0.1292	0.0013	0.0040	0.0054
9	77	0	1	0.0000	0.0186	0.0000	0.0002	0.0002
14	30	1	0	0.0473	0.0000	0.0015	0.0000	0.0015
15	26	1	1	0.0544	0.0544	0.0020	0.0020	0.0041

22.17 Answer: Let S be a set of n keywords. An algorithm to find all documents that contain at least k of these keywords is given below :

This algorithm calculates a reference count for each document identifier. A reference count of i for a document identifier d means that at least i of the keywords in S occur in the document identified by d. The algorithm maintains a list of records, each having two fields – a document identifier, and the reference count for this identifier. This list is maintained sorted on the document identifier field.

```
initialize the list $L$ to the empty list;
**for** (each keyword $c$ in $S$) **do**
**begin**
  $D$ := the list of documents identifiers corresponding to $c$;
  **for** (each document identifier $d$ in $D$) **do**
    **if** (a record $R$ with document identifier as $d$ is on list $L$) **then**
        $R.reference\_count$ := $R.reference\_count + 1$;
    **else begin**
        make a new record $R$;
        $R.document\_id$ := $d$;
        $R.reference\_count$ := 1;
        add $R$ to $L$;
    **end**;
**end**;
**for** (each record $R$ in $L$) **do**
  **if** ($R.reference\_count >= k$) **then**
    output $R$;
```

Note that execution of the second *for* statement causes the list D to “merge” with the list L. Since the lists L and D are sorted, the time taken for this merge is proportional to the sum of the lengths of the two lists. Thus the algorithm runs in time (at most) proportional to n times the sum total of the number of document identifiers corresponding to each keyword in S.