**44-560 Adv Topics in DB Systems Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exam 02 (100 points) KEY** *please print*

1. (8 pts) Consider the following transaction:

**begin**

**update employee**

**set job\_code = '500'**

**where emp\_num = '118';**

**update job**

**set job\_chg\_hour = 40,**

**job\_last\_update = '09/25/2012'**

**where job\_code = '400';**

**commit;**

The primary key of **Employee** is **emp\_num**; the primary key of **Job** is **job\_code**.

Assume the previous job-code for employee 118 was 510. For the job with job code 400, assume the previous hourly charge was 38.50 and that it was last updated on November 20, 2011.

Fill in the transaction log on your answer sheet. Add more rows if necessary. You may not need all of the rows.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TRL**  **ID** | **TRX**  **NUM** | **PREV**  **PTR** | **NEXT**  **PTR** | **OPERATION** | **TABLE** | **ROW ID** | **ATTRIBUTE** | **BEFORE**  **VALUE** | **AFTER**  **VALUE** |
| 100 | 25 | Null | 110 | START |  |  |  |  |  |
| 110 | 25 | 100 | 120 | Update | Employee | 118 | Job\_code | 510 | 500 |
| 120 | 25 | 110 | 130 | Update | Job | 400 | Job\_chg\_hour | 38.5 | 40 |
| 130 | 25 | 120 | 140 | Update | Job | 400 | Job\_last\_update | 11/20/2011 | 09/25/2012 |
| 140 | 25 | 130 | Null | Commit |  |  |  |  |  |

1. (8 pts) This question will use the sales database shown here.



Assume that

* The customer table has 5,000 rows.
* The slsrep table has 100 rows.
* The order table has 100,000 rows.
* The orderline table has 300,000 rows.
* The part table has 2,000 rows.
* 25 sales reps have commission rates greater than 0.05.
* 10 sales reps have commission rates greater than 0.08.
* 2,000 customers have a credit limit of $25,000 or more.
* There are 500 parts for which units on hand is less than 100.
* There are 30,000 rows in the orderline table for which the number ordered is > 20.

Consider the following query:

**select custNumber, custName, credLimit, slsrep.repNum, repName**

**from customer, slsrep**

**where customer.repNum = slsrep.repNum**

**and credLimit >= 25000.00;**

An access plan for executing this query is on your answer sheet. Fill in the missing information in the last four columns.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Plan** | **Step** | **Operation** | **I/O Operations** | **I/O Cost** | **Resulting Rows** | **Total I/O Cost** |
| **A** | **A1** | **Cartesian product (customer, slsrep)** | **5,000 + 100** | **5,100** | **500,000** | **5,100** |
|  | **A2** | **Select rows from A1 with matching slsrep numbers** | **500,000** | **500,000** | **5,000** | **505,100** |
|  | **A3** | **Select rows from A2 with credit limits >= 25,000** | **5,000** | **5,000** | **2,000** | **510,100** |

1. (9 pts – 2 for answer and 7 for explanation) Suppose we have a distributed database with data distributed as shown here.

**SUPPLIER(SUPPLIER\_NUMBER, CITY)** 5,000 records stored in Detroit

**PART(PART\_NUMBER, COLOR)** 50,000 records stored in Chicago

**SHIPMENT (SUPPLIER\_NUMBER, PART\_NUMBER)** 500,000 records stored in Detroit

A query is made (in SQL) to list the supplier numbers for Cleveland suppliers of red parts:

**SELECT SUPPLIER.SUPPLIER\_NUMBER**

**FROM SUPPLIER, SHIPMENT, PART**

**WHERE SUPPLIER.CITY = ‘Cleveland’**

**AND SHIPMENT.PART\_NUMBER = PART.PART\_NUMBER**

**AND SHIPMENT.SUPPLIER\_NUMBER = SUPPLIER.SUPPLIER\_NUMBER**

**AND PART.COLOR = ‘RED’;**

Each record in each relation is 100 characters long, there are twenty red parts, a history of 10,000 shipments from Cleveland, and a negligible query computation time compared with communication time. Also, there is a communication system with a data transmission time of 20,000 characters per second and two second access delay to send a message from one node to another. Query time is negligible compared to communication time.

The query access plan is to move the PART relation to Detroit, and process the entire query at the Detroit computer.

* Find the time required to do this.
* Result should be in seconds, rounded to the nearest second.
* Do *not* include the time required to send the result back from Detroit to the originating computer.

Move PART relation to Detroit:

50,000 parts \* 100 characters per part = 5,000,000 characters.

5,000,000 characters / 20,000 characters per second = 250 seconds.

250 seconds transmit time + 2 seconds access delay time = 252 seconds

Query time in Detroit is negligible so time required is 252 seconds.

The next few problems refer to the database described on the handout. A Visio diagram is provided and some sample data for some of the tables.

* Write the SQL statements described below.
* Do ***not*** use any aliases in the SQL code you write.
* Do ***not*** use more complex statements than necessary; in particular, do not use joins or group by unless necessary.
* Each clause (select, from, where, etc.) ***must*** begin on a separate line.
* Sample data is attached. ***Your code must work for any valid data***.
* For the sample output provided, the ***order of the rows is irrelevant unless you are specifically asked to produce output in a given order***.
* Each question is worth 5 points.

1. Write a single SQL statement that displays the customer code and last name for each customer with an area code of 713. For the data provided, the output of your statement should be

**10011 Dunne**

**10015 O'Brian**

**10018 Farriss**

**select cus\_code, cus\_lname**

**from customer**

**where cus\_areacode = '713';**

1. Write a single SQL statement that creates an index named cus\_index on the cus\_lname field of the customer table.

**create index cus\_index on customer(cus\_lname);**

1. Write a single SQL statement that displays the average balance of all customers with area code 713.

**select avg(cus\_balance)**

**from customer**

**where cus\_areacode = '713';**

1. Write a single SQL statement that displays the area codes appearing in the customer table. Each area code should appear only once in the list. For the data provided, the output of your statement should be

**615**

**713**

**select distinct cus\_areacode**

**from customer;**

1. Write a single SQL statement that displays the customer code, last name, and balance for each customer with an area code of 615 and with a balance greater than 200.00. Output should be sorted by customer balance. For the data provided, the output of your statement should be

**10016 Brown 221.19**

**10012 Smith 345.86**

**10013 Olowski 536.75**

**10017 Williams 768.93**

**select cus\_code, cus\_lname, cus\_balance**

**from customer**

**where cus\_areacode = '615'**

**and cus\_balance > 200.00**

**order by cus\_balance;**

1. Write a single SQL statement that displays each area code that appears in the CUSTOMER table, and the number of times it appears. For the data provided, the output of your statement should be

**615 7**

**713 3**

**select cus\_areacode, count(cus\_code)**

**from customer**

**group by cus\_areacode;**

1. Write a single SQL statement that displays each area code that appears in the CUSTOMER table, and the number of times it appears, provided the number of times is greater than 5. For the data provided, the output of your statement should be

**615 7**

**select cus\_areacode, count(cus\_code)**

**from customer**

**group by cus\_areacode**

**having count(cus\_code) > 5;**

1. Write a single SQL statement that displays the last name of each customer whose last name begins with “O” and contains at least one “i”. For the data provided, the output of your statement should be

**O'Brian**

**Olowski**

**select cus\_lname**

**from customer**

**where cus\_lname like 'O%i%';**

1. Write a single SQL statement that displays the invoice number and date for each invoice, along with the last name of the customer who placed the order. For the data provided, the output of your statement should be as shown below. Assume that the default date format is DD-MON-YY.

**1008 17-JAN-08 Dunne**

**1004 17-JAN-08 Dunne**

**1002 16-JAN-08 Dunne**

**1003 16-JAN-08 Smith**

**1006 17-JAN-08 Orlando**

**1001 16-JAN-08 Orlando**

**1007 17-JAN-08 O'Brian**

**1005 17-JAN-08 Farriss**

**select inv\_number, inv\_date, cus\_lname**

**from invoice, customer**

**where customer.cus\_code = invoice.cus\_code;**

**Multiple choice (30 points – 2 points each).**  ***Write the letter corresponding to the BEST correct answer on your answer sheet.***

***Select only ONE answer for each question. If you select more than one answer, the entire question will be counted as wrong.***

1. A transaction always consists of a single SQL statement.
   1. true
   2. false
2. Each transaction must pass the ACID test. Which of the following refers to the Atomicity property of the ACID test?
   1. data used during a transaction must be isolated from other transactions that use the same data
   2. once a transaction is completed, the changes are permanent
   3. the result of a transaction must leave the database in a consistent state
   4. all operations of the transaction must be completed
3. If two transactions are executed at the same time, the result must be the same as if they were executed one after the other. This is the same as saying that the transactions are
   1. consistent
   2. durable
   3. serializable
   4. controlled
4. In SQL, the \_\_\_\_ statement causes all changes made since the last commit statement to be aborted.
   1. rollback
   2. abort
   3. undo
   4. checkpoint
5. Suppose transaction A holds an exclusive lock on a table in a database. Transaction B wants read-only access to the same table and requests a shared lock. Will the request be granted?
   1. yes
   2. no
6. Timestamping prevents starvation.
   1. true
   2. false
7. Which of the following is/are true of the SQL cache?
   1. SQL cache is shared memory
   2. the most recently executed SQL statements are stored in the SQL cache
   3. triggers and functions are *not* stored in the cache
   4. all of the above are true
   5. only a) and b) are true
8. RAM access is much faster than hard-disk access.
   1. true
   2. false
9. Which of the following is/are true of indexes?
   1. for each insertion and deletion, all indexes must be updated
   2. for optimal performance, all columns should be indexed
   3. indexes may slow down searching
   4. all of the above are true
   5. only a) and b) are true
10. \_\_\_\_\_\_ use preset fixed-cost values for each SQL operation
    1. cost-based optimizers
    2. rule-based optimizers
11. \_\_\_\_\_ distributed databases use the same DBMS at each node.
    1. homogeneous
    2. heterogeneous
12. Which of the following is/are true of synchronous updates?
    1. data updates are immediately applied to all copies throughout the network
    2. copies of the data are always identical
    3. synchronous updates provide better data integrity than asynchronous updates
    4. all of the above are true
    5. only a) and c) are true
13. Which of the following is true of horizontal partitioning
    1. a separate copy of all or part of the database is stored at two or more sites
    2. different rows of a table are at different sites
    3. different columns of a table are at different sites
    4. all of the above are true
    5. only a) and c) are true
14. Suppose we have a transaction that references multiple database sites. However, each individual SQL statement in the transaction references only one database site. This is best described by the term
    1. remote request (level 1)
    2. remote transaction (level 2)
    3. distributed transaction (level 3)
    4. distributed request (level 4)
15. In a distributed database, there is a DBMS at each local node. There is also a DDBMS that coordinates the entire system. The DDBMS resides at only *one* of the local nodes.
    1. true
    2. false