Assignment 3

Chapter12:

1. A _____ contains the description of the entire database as seen by the database administrator.

a.distributed global dictionary
b.distributed data dictionary
c.distributed global schema
d.distributed data schema

2. A _____ distributed database system will support different database management systems (DBMS) that may even support different models running under different computer systems.

a.fully heterogeneous
b.fully homogeneous
c.homogeneous
d.heterogeneous

3. A _____ lets a single SQL statement access the data that are to be processed by a single remote database processor.

a.remote request
b.remote transaction
c.distributed request
d.distributed transaction

4. A _____ request lets a single SQL statement reference data located at several different local or remote DP sites.

a.distributed
b.transaction
c.fragmented
d.remote

5. A centralized database management is subject to a problem such as _____.

a.a growing number of remote locations
b.maintaining and operating small database systems
c.dependence on multiple sites
d.organizational flexibility of the database

6. A database management system needs _____ to prepare the data for presentation to the end user or to an application program.

a.security
b.concurrency control
c.formatting
d.I/O interface

7. A disadvantage of a distributed database management system (DDBMS) is that:

a.it is slower in terms of data access.
b.adding new sites affects other sites' operations.
c.it is processor dependent.
d.there are lack of standards.

8. A distributed _____ can reference several different local or remote data processing sites.

a.request
b.site
c.data location
d.transaction

9. A distributed database is composed of several parts known as database _____.

a.sections
b.fragments
c.partitions
d.parts

10. A(n) _____ database stores each database fragment at a single site.

a.partially replicated
b.unreplicated
c.fully replicated
d.partitioned

11. Distributed processing does not require:

a.database processing functions to be distributed to all data storage sites.
b.an existing distributed database.
c.a network of interconnected components.
d.multiple sites to share processing chores.

12. The _____ guarantees that if a portion of a transaction operation cannot be committed, all changes made at the other sites participating in the transaction will be undone to maintain a consistent database state.

a.DO-UNDO-REDO protocol
b.two-phase commit protocol (2PC)
c.coordinator protocol
d.write-ahead protocol

13. The _____ processor is the software component found in each computer that requests data. It receives and processes the application's data requests.

a.database
b.transaction
c.data
d.network

14. The _____ rule requires that all copies of data fragments be identical.
a.shared fragment
b.mutual consistency
c.horizontal fragmentation
d.replication

15. The objective of _____ optimization is to minimize the total cost associated with the execution of a request.
a.data
b.remote
c.transaction
d.query

16. _____ distributed database management system (DDBMS) integrate multiple instances of the same DBMS over a network.
a.Homogeneous
b.Heterogeneous
c.Fully heterogeneous
d.Combination

17. _____ fragmentation allows a user to break a single object into two or more segments, or fragments.
a.Horizontal
b.Vertical
c.Data
d.Request

18. _____ fragmentation refers to the division of a relation into attribute subsets.
a.Data
b.Horizontal
c.Vertical
d.Mixed

19. _____ fragmentation refers to the division of a relation into subsets of tuples.
a.Vertical
b.Horizontal
c.Data
d.Mixed

20. _____ is the delay imposed by the amount of time required for a data packet to make a round trip from point A to point B.
a.Data distribution
b.Replica transparency
c.Network latency
d.Network partitioning

21. _____ transparency allows a physically dispersed database to be managed as though it were centralized.
a.Distribution
b.Transaction
c.Failure
d.Performance

22. _____ transparency allows data to be updated simultaneously at several network sites.
a.Transaction
b.Distribution
c.Failure
d.Performance

23. _____ transparency allows the system to operate as if it were a centralized database management system.
a.Heterogeneity
b.Distribution
c.Performance
d.Failure

24. _____ transparency exists when the end user or programmer must specify the database fragment names but does not need to specify where these fragments are located.
a.Transaction
b.Location
c.Local mapping
d.Fragmentation

25. _____ transparency is the highest level of transparency. The end user or programmer does not need to know that a database is partitioned.
a.Performance
b.Fragmentation
c.Location
d.Local mapping

Deliverables

RemoteLabs Oracle Server
http://labs.cci.fsu.edu/

1. Log into **Oracle** Server using RemoteLabs:
   Notes > **Oracle SQL Developer Login** (see **helper video**, along with displayed instructions)

2. Create and populate **Oracle** tables: http://www.qcitr.com/vids/LIS3781_A3_Oracle_EStore.mp4
   (Must include same number of tables and records for each table as illustrated in the above video.)

Errata (corrected text in video)

3. **Note:** Time (approx. 25:00): should be seven *total* digits, *not* nine.

Deliverables

1. **Tables and data: Using only SQL: Save as lis3781_a3_solutions.sql**
2. **SQL solutions (below)**

No Credit will be given if tables and data are not populated in RemoteLabs Oracle Server.

Note:

README.md file should include the following items:
1. Screenshot of ***your*** SQL code;
2. Screenshot of ***your*** populated tables (w/in the **Oracle environment**);
3. Optional: SQL code for the required reports.
4. Bitbucket repo links: ***Your*** lis3781 Bitbucket repo link

Deliverables (see screenshots below):

1. Provide **Bitbucket** read-only access to **course** repo, using Markdown syntax, (**README.md** must also include screenshots per above.)
   (**DO NOT create README in Bitbucket**—ALWAYS do it locally, then push it to Bitbucket.)
2. **FSU's Learning Management System**: include course **Bitbucket** repo link

Reports

The following items are *required* (use RemoteLabs – Oracle Server):

NOTE: All dollar amounts must be formatted to two decimal places, including a dollar sign ($). All phone numbers and zip codes must include proper hyphens (-).
Note: Results to text **F5**, results to grid **F9** (must select SQL first!)
Note: Change text size: **Tools > Preferences > Code Editor > Fonts**

1. Display Oracle version (one method).
2. Display Oracle version (another method).
3. Display current user.
4. Display current day/time (formatted, and displaying AM/PM).
5. Display your privileges.
6. Display all user tables.
7. Display structure for each table.
8. List the customer number, last name, first name, and e-mail of every customer.
9. Same query as above, include street, city, state, and sort by state in descending order, and last name in ascending order.
10. What is the full name of customer number 3? Display last name first.
11. Find the customer number, last name, first name, and current balance for every customer whose balance exceeds $1,000, sorted by largest to smallest balances.
12. List the name of every commodity, and its price (formatted to two decimal places, displaying $ sign), sorted by smallest to largest price.

13. Display all customers' first and last names, streets, cities, states, and zip codes as follows (ordered by zip code descending).

```
NAME                                            ADDRESS
Victors, Sally                                  534 Holler Way, Charleston, WV 78345
Taylor, Stephen                                 456 Elm St., St. Louis, MO 57252
Davis, Beverly                                  123 Main St., Detroit, MI 48252
Carter, Donna                                   789 Peach Ave., Los Angeles, CA 48252
Silverman, Robert                               857 Wilbur Rd., Phoenix, AZ 25278
```

14. List all orders not including **cereal**--use subquery to find commodity id for cereal.
15. List the customer number, last name, first name, and balance for every customer whose balance is between $500 and $1,000, (format currency to two decimal places, displaying $ sign).
16. List the customer number, last name, first name, and balance for every customer whose balance is greater than the average balance, (format currency to two decimal places, displaying $ sign).
17. List the customer number, name, and ***total*** order amount for each customer sorted in descending ***total*** order amount, (format currency to two decimal places, displaying $ sign), and include an alias "total orders" for the derived attribute.
18. List the customer number, last name, first name, and complete address of every customer who lives on a street with "**Peach**" anywhere in the street name.
19. List the customer number, name, and ***total*** order amount for each customer whose ***total*** order amount is greater than $1500, for each customer sorted in descending ***total*** order amount, (format currency to two decimal places, displaying $ sign), and include an alias "total orders" for the derived attribute.
20. List the customer number, name, and number of units ordered for orders with 30, 40, or 50 units ordered.
21. Using EXISTS operator: List customer number, name, number of orders, minimum, maximum, and sum of their order total cost, only if there are 5 or more customers in the customer table, (format currency to two decimal places, displaying $ sign).
22. Find aggregate values for customers:
    (Note, difference between count(*) and count(cus_balance), one customer does not have a balance.)
23. Find the number of unique customers who have orders.
24. List the customer number, name, commodity name, order number, and order amount for each customer order, sorted in descending order amount, (format currency to two decimal places, displaying $ sign), and include an alias "order amount" for the derived attribute.
25. Modify prices for DVD players to $99.
    Note: First, *be sure* to **SET DEFINE OFF** (don't use a semi-colon on the end).

References

1. http://www.sqlcourse.com/
2. https://www.oracle-dba-online.com/sql/oracle_sql_tutorial.htm
3. https://www.w3schools.com/sql/
4. https://www.tutorialspoint.com/oracle_sql/index.asp
5. http://comet.lehman.cuny.edu/jung/cmp420758/OracleTutorial.pdf
6. **Format Models**: https://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm
7. https://www.mssqltips.com/sqlservertip/2944/comparing-sql-server-and-oracle-datatypes/
8. **SQL Server Data Types**:
   https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017

9. **Oracle Data Types**:
   https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements001.htm#SQLRF0021
   https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#CNCPT1832
   https://www.techonthenet.com/oracle/datatypes.php
   https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#CNCPT1832

Note: number data type
For numeric columns, you can specify the column as:

```
column_name NUMBER
```

Optionally, you can also specify a **precision** (total number of digits) and **scale** (number of digits to the right of the decimal point):

```
column_name NUMBER (precision, scale)
```

If a precision is not specified, the column stores values as given. If no scale is specified, the scale is zero.
Oracle guarantees portability of numbers with a precision equal to or less than 38 digits. You can specify a scale and no precision:

```
column_name NUMBER (*, scale)
```

In this case, the precision is 38, and the specified scale is maintained.
When you specify numeric fields, it is a good idea to specify the precision and scale. This provides extra integrity checking on input.

Table 26-1 How Scale Factors Affect Numeric Data Storage

Input Data	Specified As	Stored As
7,456,123.89	`NUMBER`	7456123.89
7,456,123.89	`NUMBER(*,1)`	7456123.9
7,456,123.89	`NUMBER(9)`	7456124
7,456,123.89	`NUMBER(9,2)`	7456123.89
7,456,123.89	`NUMBER(9,1)`	7456123.9
7,456,123.89	`NUMBER(6)`	(not accepted, exceeds precision)
7,456,123.89	`NUMBER(7,-2)`	7456100

If you specify a negative scale, Oracle Database rounds the actual data to the specified number of places to the left of the decimal point. For example, specifying (7,-2) means Oracle Database rounds to the nearest hundredths, as shown in Table 26-1.