Assignment 2

Chapter11:

1. A system table space, a user data table space, an index table space, and a temporary table space are examples of _____.

a.procedure caches
b.file groups
c.data caches
d.operation modes

2. A(n) _____ is a logical grouping of several data files that store data with similar characteristics.

a.procedure cache
b.table space
c.data cache
d.listener

3. Automatic query optimization means that the:

a.optimization takes place at compilation time by the programmer.
b.DBMS finds the most cost-effective access path without user intervention.
c.optimization process is scheduled and selected by the end user or programmer.
d.database access strategy is defined when the program is executed.

4. From the performance point of view, _____ databases eliminate disk access bottlenecks.

a.RAID
b.distributed
c.index-organized
d.in-memory

5. If there is no index, the DBMS will perform a _____ scan.

a.loop
b.range
c.row ID table access
d.full table

6. In standard SQL, the optimizer hint ALL_ROWS is generally used for _____ mode processes.

a.interactive
b.real-time
c.batch
d.transaction

7. In the context of RAID levels, _____ refers to writing the same data blocks to separate drives.

a.striping
b.mirroring
c.partitioning
d.aggregating

8. Knowing the sparsity of a column helps you decide whether the use of _____ is appropriate.

a.query processing
b.query optimization
c.an index
d.a full table scan

9. On the client side, the objective is to generate an SQL query that returns a correct answer in the least amount of time, using a minimum amount of resources at the server end. The activities required to achieve this goal are commonly referred to as _____ tuning.

a.client SQL
b.database SQL
c.SQL performance
d.DBMS performance

10. On the server side, the database environment must be properly configured to respond to clients' requests in the fastest way possible, while making optimum use of existing resources. The activities required to achieve this goal are commonly referred to as _____ tuning.

a.client and server
b.database SQL
c.SQL performance
d.DBMS performance

11. The DBMS _____ the SQL query and chooses the most efficient access/execution plan.

a.parses
b.executes
c.fetches
d.processes

12. The DBMS _____ the data and sends the result set back to the client.

a.parses
b.executes
c.fetches
d.processes

13. The _____ cache is used as a temporary storage area for ORDER BY or GROUP BY operations, as well as for index-creation functions.

a.data
b.SQL
c.sort
d.optimizer

14. The _____ is a shared, reserved memory area that stores the most recently executed SQL statements or PL/SQL procedures, including triggers and functions.

a.buffer cache
b.procedure cache
c.data cache
d.permanent storage

15. The _____ must be set large enough to permit as many data requests to be serviced from cache as possible.

a.data cache
b.SQL cache
c.sort cache
d.optimizer mode

16. The _____ process analyzes SQL queries and finds the most efficient way to access data.

a.optimizer
b.scheduler
c.listener
d.user

17. The _____ table space is used to store the data dictionary tables.
a.system
b.user data
c.temporary
d.rollback segment

18. The data cache is where the data read from the database data files are stored _____ the data have been read or _____ the data are written to the database data files.
a.after; before
b.after; after
c.before; before
d.before; after

19. The data cache or _____ is a shared, reserved memory area that stores the most recently accessed data blocks in RAM.
a.buffer cache
b.procedure cache
c.SQL cache
d.permanent storage

20. The majority of primary memory resources will be allocated to the _____ cache.
a.data
b.SQL
c.sort
d.optimizer

21. To work with data, a DBMS must retrieve the data from _____ and place them in _____.
a.data files; procedure cache
b.RAM; data cache
c.permanent storage; RAM
d.temporary files; procedure cache

22. When moving data from permanent storage to RAM, an I/O disk operation retrieves:
a.an entire table.
b.an entire physical disk block.
c.only the row containing the attribute requested.
d.only the attribute which was requested.

23. When setting optimizer hints, _____ instructs the optimizer to minimize the overall execution time, that is, to minimize the time it takes to return the total number of rows in the query result set. This hint is generally used for batch mode processes.
a.ALL_ROWS
b.FIRST_ROWS
c.INDEX(P_QOH_NDX)
d.OPTIMIZATION_ROWS

24. Which of the following is the first step of query processing at the DBMS server end?
a.Executing
b.Parsing
c.Fetching
d.Delivering

25. _____ is the central activity during the parsing phase in query processing.
a.Clustering
b.Partitioning
c.Query validation
d.Query optimization

26. _____ refers to the number of different values a column could possibly have.
a.Database statistics
b.Data sparsity
c.A bitmap index
d.Clustering

Deliverables
Part 1

Distributed Version Control with Git and Bitbucket, and Development Environments

Carefully go through the following steps: http://www.qcitr.com/usefullinks.htm#lesson3b

Part 2

MySQL Server

Using only SQL: Save as lis3781_a2_solutions.sql

A. Tables and insert statements.
B. Include indexes and foreign key SQL statements (see below).
C. Include ***your*** query result sets, including grant statements.
D. The following tables should be created and populated with at least 5 records **both** locally and to the CCI server.
E. **No Credit will be given if tables and data do not forward-engineer to the CCI server.**

1. **Using SQL ONLY**, **NOT** MySQL Workbench:
2. Locally: create **yourfsuid** database, and two tables: **company** and **customer**
   NOTE: Also, these two tables must be populated in **yourfsuid** database on the CCI server.
   a. Use 1:M relationship: **company** is parent table
   b. **company** attributes:
      i. cmp_id (pk)
      ii. cmp_type enum('C-Corp','S-Corp','Non-Profit-Corp','LLC','Partnership')
      iii. cmp_street
      iv. cmp_city
      v. cmp_state
      vi. cmp_zip (zf)
      vii. cmp_phone
      viii. cmp_ytd_sales
      ix. cmp_url
      x. cmp_notes
   c. **customer** attributes:
      i. cus_id (pk)
      ii. cmp_id (fk)
      iii. cus_ssn (binary 64)
      iv. cus_salt (binary 64)
      v. cus_type enum('Loyal','Discount','Impulse','Need-Based','Wandering')
      vi. cus_first
      vii. cus_last
      viii. cus_street
      ix. cus_city
      x. cus_state
      xi. cus_zip (zf)
      xii. cus_phone
      xiii. cus_email
      xiv. cus_balance
      xv. cus_tot_sales
      xvi. cus_notes
   d. Create suitable indexes **and** foreign keys:
      (Review Notes > **Enforcing_PK_FK_Relationship.pdf**)
   e. Enforce pk/fk relationship: on update cascade, on delete restrict

Local Admin: (***NOTE***:** you do ***not*** have permissions to create users on the CCI Server)

Create two different users (user3 and user4), with two different passwords: both users can access from **localhost** only. (**Note:** user1 and user2 should have already been created from the practice ex.)

See example:Notes>DBA>**Granting_Privileges.pdf**

1. Limit **user3** to select, update, and delete privileges on company and customer tables
2. Limit **user4** to select, and insert privileges on customer table

Log into local server as each user:

3. Verify database/table permissions, show grants:
    a. you/admin
    b. user3
    c. user4
4. Display current user (**user4**) and MySQL version
5. List tables (as admin)
6. Display structures for both tables
    a. company
    b. customer
7. Display data for both tables:
    a. company
    b. customer
8. **Verify fk options:** Display query result set of customer table, including modified fk, by updating pk in parent table (company), change pk value from 1 to 6. Copy and paste SQL commands and query result sets displaying change:
9. **Verify fk options:** Display the SQL statement(s), and query result set that prevented the parent table (company) from deleting a record w/o deleting the associated child table (customer) records first. Include delete statement, and resulting error.
10. Log in as **user3**:
    a. show the SQL INSERT statement, **and** corresponding query result set that prevented user3 from inserting data in the company table
    b. show the SQL INSERT statement, **and** corresponding query result set that prevented user3 from inserting data in the customer table
11. Log in as **user4**:
    a. show the SQL statement, **and** corresponding query result set that prevented user4 from "seeing" company table:
    b. same as above, though, prevented from being able to delete from the customer table:
12. Log in as **admin**: remove both tables (structure and data), and show commands:
13. **NOTE:** ***MUST*** include Bitbucket repo link.

Note:

README.md file should include the following items:

1. Screenshot of ***your*** SQL code;
2. Screenshot of ***your*** populated tables;
3. git commands w/short descriptions;
4. Bitbucket repo links:
    a. ***Your*** lis3781 Bitbucket repo link
    b. The completed tutorial repo above (**bitbucketstationlocations**). (See link in screenshot below.)

Deliverables (see screenshots below):

1. Provide **Bitbucket** read-only access to **lis3781** repo, include links to the repo (**BitbucketStationLocations**) you created in the above tutorials in **README.md**, using Markdown syntax, (**README.md** must also include screenshots per above.)
2. **FSU's Learning Management System**: include lis4368 **Bitbucket** repo link

*****Examples*****

LIS3781_A2a.png

```
/*
A character set is a set of symbols and encodings.
A collation is a set of rules for comparing characters in a character set.

Suppose that we have an alphabet with four letters: "A", "B", "a", "b".
We give each letter a number: "A" = 0, "B" = 1, "a" = 2, "b" = 3.
The letter "A" is a symbol, the number 0 is the encoding for "A",
and the combination of all four letters and their encodings is a character set.

Suppose that we want to compare two string values, "A" and "B".
The simplest way to do this is to look at the encodings: 0 for "A" and 1 for "B".
Because 0 is less than 1, we say "A" is less than "B". What we've just done is apply a collation to our character set.
The collation is a set of rules (only one rule in this case): "compare the encodings."
We call this simplest of all possible collations a binary collation.

http://dev.mysql.com/doc/refman/5.5/en/charset.html
*/

-- set foreign_key_checks=0;

drop database if exists mjowett;
create database if not exists mjowett;
use mjowett;

-- -----------------------------------------------------
-- Table company
-- -----------------------------------------------------
DROP TABLE IF EXISTS company;
CREATE  TABLE IF NOT EXISTS company
(
  cmp_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  cmp_type enum('C-Corp','S-Corp','Non-Profit-Corp','LLC','Partnership'),
  cmp_street VARCHAR(30) NOT NULL,
  cmp_city VARCHAR(30) NOT NULL,
  cmp_state CHAR(2) NOT NULL,
  cmp_zip int(9) unsigned ZEROFILL NOT NULL COMMENT 'no dashes',
  cmp_phone bigint unsigned NOT NULL COMMENT 'ssn and zip codes can be zero-filled, but not US area codes',
  cmp_ytd_sales DECIMAL(10,2) unsigned NOT NULL COMMENT '12,345,678.90',
  cmp_email VARCHAR(100) NULL,
  cmp_url VARCHAR(100) NULL,
  cmp_notes VARCHAR(255) NULL,
  PRIMARY KEY (cmp_id)
)
ENGINE = InnoDB CHARACTER SET utf8 COLLATE utf8_general_ci;

SHOW WARNINGS;

INSERT INTO company
VALUES
(null,'C-Corp','507 - 20th Ave. E. Apt. 2A','Seattle','WA','081226749','2065559857','12345678.00',null,'http://www.http://technologies.ci.fsu.edu/node/72','company notes1'),
(null,'S-Corp','908 W. Capital Way','Tacoma','WA','004011298','2065559482','9945678.00',null,'http://www.qcitr.com','company notes2'),
(null,'Non-Profit-Corp','722 Moss Bay Blvd.','Kirkland','WA','000337845','2065553412','1345678.00',null,'http://www.markjowett.com','company notes3'),
(null,'LLC','4110 Old Redmond Rd.','Redmond','WA','000029021','2065558122','678345.00',null,'http://www.thejowetts.com','company notes4'),
(null,'Partnership','4726 - 11th Ave. N.E.','Seattle','WA','001051082','2065551189','345678.00',null,'http://www.qualityinstruction.com','company notes5');

SHOW WARNINGS;
```

LIS3781_A2b.png

```sql

-- ----------------------------------------------------
-- Table customer
-- ----------------------------------------------------
DROP TABLE IF EXISTS customer;
CREATE  TABLE IF NOT EXISTS customer
(
  cus_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  cmp_id INT UNSIGNED NOT NULL,
  cus_ssn binary(64) not null,
  cus_salt binary(64) not null COMMENT '*only* demo purposes - do *NOT* use *salt* in the name!',
  cus_type enum('Loyal','Discount','Impulse','Need-Based','Wandering'),
  cus_first VARCHAR(15) NOT NULL,
  cus_last VARCHAR(30) NOT NULL,
  cus_street VARCHAR(30) NULL,
  cus_city VARCHAR(30) NULL,
  cus_state CHAR(2) NULL,
  cus_zip int(9) unsigned ZEROFILL NULL,
  cus_phone bigint unsigned  NOT NULL COMMENT 'ssn and zip codes can be zero-filled, but not US area codes',
  cus_email VARCHAR(100) NULL,
  cus_balance DECIMAL(7,2) unsigned NULL COMMENT '12,345.67',
  cus_tot_sales DECIMAL(7,2) unsigned NULL,
  cus_notes VARCHAR(255) NULL,
  PRIMARY KEY (cus_id),

  UNIQUE INDEX ux_cus_ssn (cus_ssn ASC),
  INDEX idx_cmp_id (cmp_id ASC),

/*
Comment CONSTRAINT line to demo DBMS auto value when *not* using "constraint" option for foreign keys, then...
SHOW CREATE TABLE customer;
*/
 CONSTRAINT fk_customer_company
   FOREIGN KEY (cmp_id)
   REFERENCES company (cmp_id)
   ON DELETE NO ACTION
   ON UPDATE CASCADE
)
ENGINE = InnoDB CHARACTER SET utf8 COLLATE utf8_general_ci;

SHOW WARNINGS;

-- salting and hashing sensitive data (e.g., SSN). Normally, *each* record would receive unique random salt!
set @salt=RANDOM_BYTES(64);

INSERT INTO customer
VALUES
(null,2,unhex(SHA2(CONCAT(@salt, 000456789),512)),@salt,'Discount','Wilbur','Denaway','23 Billings Gate','El Paso','TX','085703412','2145559857','test1@mymail.com','8391.87','37642.00','customer notes1'),
(null,4,unhex(SHA2(CONCAT(@salt, 001456789),512)),@salt,'Loyal','Bradford','Casis','891 Drift Dr.','Stanton','TX','005819045','2145559482','test2@mymail.com','675.57','87341.00','customer notes2'),
(null,3,unhex(SHA2(CONCAT(@salt, 002456789),512)),@salt,'Impulse','Valerie','Lieblong','421 Calamari Vista','Odessa','TX','000621134','2145553412','test3@mymail.com','8730.23','92678.00','customer notes3'),
(null,5,unhex(SHA2(CONCAT(@salt, 003456789),512)),@salt,'Need-Based','Kathy','Jeffries','915 Drive Past','Penwell','TX','009135674','2145558122','test4@mymail.com','2651.19','78345.00','customer notes4'),
(null,1,unhex(SHA2(CONCAT(@salt, 004456789),512)),@salt,'Wandering','Steve','Rogers','329 Volume Ave.','Tarzan','TX','000054426','2145551189','test5@mymail.com','782.73','23471.00','customer notes5');

SHOW WARNINGS;
-- set foreign_key_checks=1;

select * from company;
select * from customer;
```

LIS3781 A2 Populated Tables

```
c:\mysql\bin>mysql -u mjowett -p --port 3308
Enter password: *******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.19 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mjowett
Database changed
mysql> select * from company;
+--------+------------------+---------------------------+----------+-----------+-----------+------------+---------------+-----------+-------------------------------------------------+----------------+
| cmp_id | cmp_type         | cmp_street                | cmp_city | cmp_state | cmp_zip   | cmp_phone  | cmp_ytd_sales | cmp_email | cmp_url                                         | cmp_notes      |
+--------+------------------+---------------------------+----------+-----------+-----------+------------+---------------+-----------+-------------------------------------------------+----------------+
|      1 | C-Corp           | 507 - 20th Ave. E. Apt. 2A | Seattle  | WA        | 081226749 | 2065559857 |   12345678.00 | NULL      | http://www.http://technologies.ci.fsu.edu/node/72 | company notes1 |
|      2 | S-Corp           | 908 W. Capital Way        | Tacoma   | WA        | 004011298 | 2065559482 |    9945678.00 | NULL      | http://www.qcitr.com                            | company notes2 |
|      3 | Non-Profit-Corp  | 722 Moss Bay Blvd.        | Kirkland | WA        | 000337845 | 2065553412 |    1345678.00 | NULL      | http://www.markjowett.com                       | company notes3 |
|      4 | LLC              | 4110 Old Redmond Rd.      | Redmond  | WA        | 000029021 | 2065558122 |     678345.00 | NULL      | http://www.thejowetts.com                       | company notes4 |
|      5 | Partnership      | 4726 - 11th Ave. N.E.     | Seattle  | WA        | 001051082 | 2065551189 |     345678.00 | NULL      | http://www.qualityinstruction.com               | company notes5 |
+--------+------------------+---------------------------+----------+-----------+-----------+------------+---------------+-----------+-------------------------------------------------+----------------+
5 rows in set (0.00 sec)

mysql> select * from customer;
+--------+--------+-----------------------------------------------+----------------------------------------------------+------------+-----------+----------+--------------------
| cus_id | cmp_id | cus_ssn                                       | cus_salt                                           | cus_type   | cus_first | cus_last | cus_street
+--------+--------+-----------------------------------------------+----------------------------------------------------+------------+-----------+----------+--------------------
|      1 |      2 | ß%¤³~┐Ê|³▒_V<▼Ô1îPG4¾━ûC 6■Ë♠»±e☻Ac‖W/È5ëgß╓Ñ×  -uìäÛÞ|UòB?[╩ | ♂→Í¶Ë♣!u5▯¼ý]↕º¥*█  ♠▯¡├¶r╨ß♦5└Û►ª╬ôl©èÎZo┤–³☻2▯►Õ]C²▀ux↓v{h+è♀ | Discount   | Wilbur    | Denaway  | 23 Billings Gate
c35óû8NÔCGX±┴6cPñÂ♣Î|♫└.¾═C-ksf½c ïnòåùÇNH← M®Ç♫
╟h²Y◄üo♣┌u | ♂→Í¶Ë♣!u5▯¼ý]↕º¥*█ ♠▯¡├¶r╨ß♦5└Û►ª╬ôl©èÎZo┤–³☻2▯►Õ]C²▀ux↓v{h+è♀ | Loyal      | Bradford  | Casis    | 891 Drift Dr.      | Stanton  | TX        | 005819045 | 2145559482 | test2@mymail.com |
╚®l˚HXwÍ╓RÖè˚‖³←½öÓG¥Vì÷? | ♂→Í¶Ë♣!u5▯¼ý]↕º¥*█  ♠▯¡├¶r╨ß♦5└Û►ª╬ôl©èÎZo┤–³☻2▯►Õ]C²▀ux↓v{h+è♀ | Impulse    | Valerie   | Lieblong | 421 Calamari Vista | Odessa   | TX        | 000621134 | 2145553412 | test
|      4 |      5 | 9üÂ#Ò!0Õ♦ÛbR█↔ôÀ■ÿ  .2:Fà↔±áô!└ÏÄ┤¨º+Ï¤Y×EÈkí▦éýÀ¬ë§Å┬n¯☻¤ká♥ZäF | ♂→Í¶Ë♣!u5▯¼ý]↕º¥*█        ♠▯¡├¶r╨ß♦5└Û►ª╬ôl©èÎZo┤–³☻2▯►Õ]C²▀ux↓v{h+è♀ | Need-Based | Kathy      | Jeffries | 915 Drive
|      5 |      1 | §|Ø☺˚"R8Õ¼M┴±«W¶¬
¾w╓◄©■━hx|-<Ý↓_↓↕.b╗zkÃ&ï®LZ µC ▼<¯å5³‼ÄâJ♠Î½☻ | ♂→Í¶Ë♣!u5▯¼ý]↕º¥*█      ♠▯¡├¶r╨ß♦5└Û►ª╬ôl©èÎZo┤–³☻2▯►Õ]C²▀ux↓v{h+è♀ | Wandering  | Steve     | Rogers   | 329 Volume Ave.    | Tarzan   | TX        | 0000
+--------+--------+-----------------------------------------------+----------------------------------------------------+------------+-----------+----------+--------------------
5 rows in set (0.00 sec)

mysql>
```