

Database Systems

Lab # 3

Objectives

1. Using Mysql Workbench Data modeler tool to model and build a relational schema.
2. Learning how to manage database memory.
3. Learning how to manage database storage.
4. Learning how to manage users and security.
5. Introducing Stored Procedures as a powerful tool to validate data entering and reading from the database.

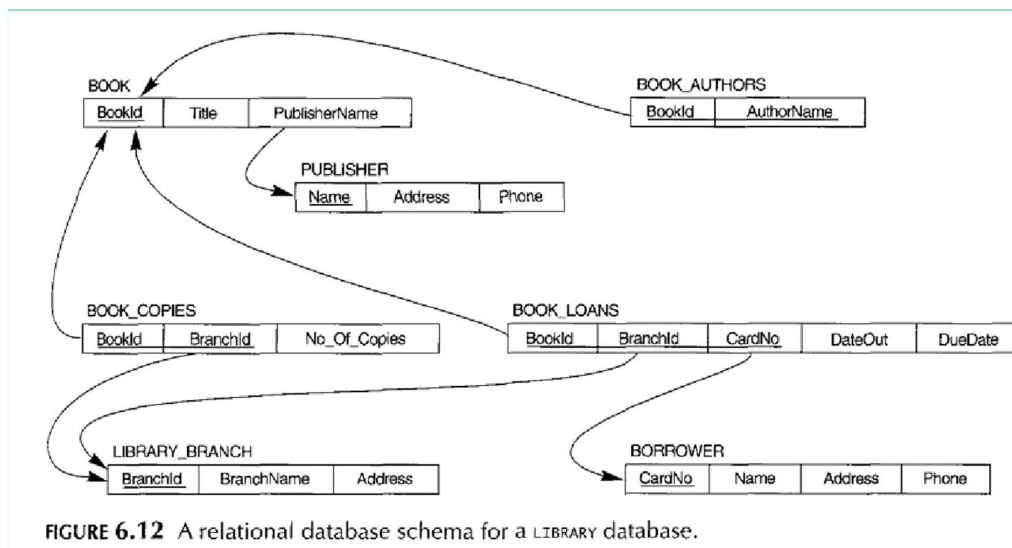
References

- [1] R.Elmasri and S. Navathe, Fundamentals of Database Systems, 6th edition
[2] <http://dev.mysql.com/doc/refman/5.7/en/>

Reading

<https://dev.mysql.com/doc/workbench/en/wb-data-modeling.html>
<https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-creating-a-model.html>
<https://www.mysql.com/products/workbench/admin/>
<https://dev.mysql.com/doc/refman/5.7/en/sql-syntax-server-administration.html>
<https://dev.mysql.com/doc/refman/5.7/en/connecting.html>

Lab



Data Modeling and Schema Generation

1. Use the data modeler module in the Workbench
2. Create a new schema; call it *SAMPLEx* where *x* is your seat number.
3. Draw the ERD for figure 6.14 in Reference [1] (figure 6.12 in 4th edition), as shown in the schema figure section.
4. Generate the database tables in the schema and the creation DDL scripts.
5. Write the steps and show the ERD, DDL scripts, created tables.
6. Connect to the database and verify the creation of tables.
7. Save the generated scripts to your hard disk.

Database Administration

8. Using the SQL Developer, create a new user account called *SAMPLE* and grant it as *DBA*.
9. Write a simple program or use an offline tool, to insert about 200,000

records or more for each table

10. Using the SAMPLE account, report the storage details about: data files, free space, offline table spaces, table space quotas, and table space segments before and after the insertion.
11. Compact the database tables storage, and report whether it recovered the unused spaces or not.
12. establish a local network between your database host machine, and another machine that will be a database client. Then, configure the network environment between the two machines and connect from the client machine to the database on the host one.
13. Select the publishers that have phone numbers ended with '39' and show this result.
14. Expire the password of the HR account and drop the SAMPLE one.

Stored Procedures

15. Consider the following database schema:

DEPT (Dnumber , Dname, Founded, Mgr_ssn, Budget)

EMPLOYEE (Ssn, Ename , Bdate, Dno, Salary)

Note that the attribute Founded represents the foundation date of the department. Create the database schema containing tables above.

16. Create a stored function Count_Emp (Dnumber NUMBER) that returns the number of employees working for the department Dnumber
17. Create a stored procedure that ensures that Year(DEPT.Founded) >=1960 for all departments; if a row violates this constraint then set its date to be '01-JAN-1960'
18. Create a trigger to ensure that no department has more than 8 employees
19. Create a trigger to implement "ON UPDATE CASCADE" for the foreign key EMPLOYEE.Dno.
20. Create a trigger to ensure that whenever an employee is given a raise in salary, his department manager's salary must be increased to be at least as much.
21. Test the created functions/procedures/triggers by suitable SQL statements to ensure their correctness.

Notes:

1. For point #20 it is not possible in MySQL as it rejects updates to the same table that causes the trigger so you can limit the update for the table using a stored procedure that has the update statement then update statement that is editing employee or manager salary (do not use trigger) [for completeness only next part is not required but worth to note we could make a trigger that rejects the update or constraint on the salary field itself using assertion ,the extra trigger or constraint that is fixing a possible backdoor to violate the salary rule is completely optional, only you need to deliver stored procedure as a solution to fifth requirement. ,a possible solution also is to make the trigger cancel the raise of the salary if it defies the constraint
2. check this link <https://dev.mysql.com/doc/workbench/en/wb-performance.html> to know more about performance schema in mysql
3. Regarding reporting storage details try to map these definitions from their domain I mean these names may be used in oracle DBMS like the analogous concept of cmd commands in windows and its counterpart in Linux like "dir" vs "ls -l" but for example, you could assume table space term is analogous to buffers allocated in this query `SELECT * FROM sys.x$schema_table_statistics_with_buffer;` hint performance schemas uses views of schema named sys. you can see the different views sys schemas support and assume reasonable assumptions to map the definitions, regarding compacting table spaces you could use sql command to optimize table space and if space didn't change this is normal this may happen because you didn't insert enough data to make file fragmentation inefficiencies appears and this is completely acceptable.
4. For remote connection part , you need at least to connect mysql client to your local machine IP i.e you could use same machine and calling it through its local network ip not 127.0.0.1 , another option is to have another sql server on different local ip and connect it from a client. No need to complicate the task and make client and server on two different local networks as this will need static public ip of the server and port forwarding rules in router.