**BCDE214 Database Administration Assignment 2– Portfolio**

**Appendix B Backup And Security**

**Due Date:** Friday 2 October; 11:59 p.m. **(Total Marks 25)**

**Write scripts** to perform the following tasks

**BackUp and Data Export (Marks 2.5 \* 2 = 5)**

Write scripts to implement the following backup and data export policies:

* The data base should be fully backed up at 11:00p.m. every Monday night. The backups should expire after three weeks. There should also be a differential backup at a useful time before a Thursday auction would commence.
* The relevant data for an auction should also be exported as three csv files. These should be the clients (including contact details) who actually bought or sold stock, and the details of the auctions, including names of buyers, seller, agents and auctioneers.

**Users and Security (Marks 20)**

* Write a script to create the following SQL users with the provided rights. They should not be able to do anything else. Make the passwords the same as the Login name and do not enforce the password policy. **(Marks 1 \* 8 =8)**

|  |  |
| --- | --- |
| **User** | **Rights** |
| AuctionCreator | Load and edit data into the AuctionDay, CattleAuction and SheepAuction tables |
| ClientLoader | Load and edit data in the AuctionClient and AuctionClientAtAuction tables |
| StockLoader | Load and edit data in the \*Lot tables |
| AuctionDayDataEntry | Update data in the \*Lot tables, but not create new records |
| AuctionDaySupervisor | Everything that the ClientLoader, StockLoader and AuctionDayDataEntry user can do. |
| SaleDayAdmin | Full administrator rights. |
| Reporter | Read data from all the tables, but not update it. |
| DailyReporter | Read data from the reporting views from Appendix A. |

* Create application roles equivalent to the ClientLoader, AuctionDayDataEntry and Reporter users. **(Marks 2 \* 3 =6)**
* Give the appropriate users the rights to run the stored procedures from Appendix A.

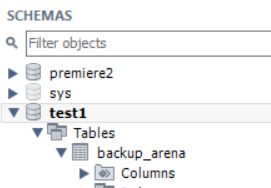
**(Marks 3)**

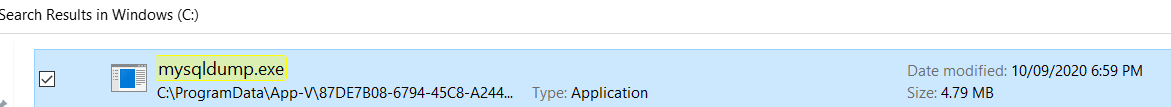
* Create a user who will rebuild the indices you created in Appendix A, and perform backup and restore functions. **(Marks 3)**

**Lesson Plan**

**Common mistakes while MySQLDump command**

**How to implement MySQL Dump Command at Ara**





##### "mysqldump" Utility

"mysqldump" can be used to back up the entire server, selected databases, or selected tables of a database. It generates a text file of SQL statements that can later be executed to recreate the database/table and its contents.

-- Dump a particular database (all tables)

Shell> mysqldump -u username -p databaseName > backupFile.sql

-- Dump selected tables of a particular database

Shell> mysqldump -u username -p databaseName table1Name [table2Name ...] > backupFile.sql

-- Dump several databases with --database option

shell> mysqldump --databases database1Name [database2Name ...] > backupFile.sql

-- Dump all databases in the server with --all-databases option, except mysql.user table (for security)

shell> mysqldump -u root -p --all-databases --ignore-table=mysql.user > backupServer.sql

You can restore from the backup by running client "mysql" in batch mode to execute the SQL statements:

Shell> mysql -u username -p databaseName < backupFile.sql

or using the source command in an interactive client:

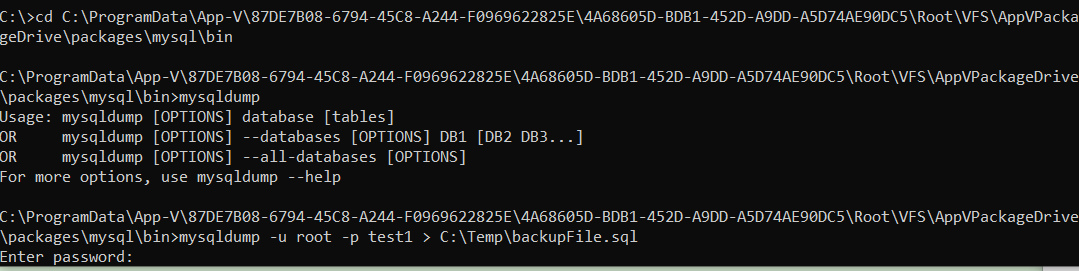
-- Run an mysql interactive client and load in to MySQL server

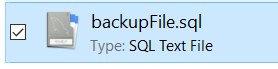
mysql> source d:/path/backupfile.sql

**Example:** The following command backup the table employees and customers of the database southwind\_mini.

Shell> mysqldump -u username -p southwind\_mini employees customers > "d:\path\backup.sql"

Study the output file, which contains CREATE TABLE and INSERT statements to recreate the tables dumped.





C:\Temp|backupFile.sql

#### 4.8  Events

An event is a set of stored SQL statements that get executed at the scheduled date and time. Once an event is completed, it will be dropped automatically.

**Syntax:**

CREATE EVENT eventName ON SCHEDULE

AT timestamp [+ INTERVAL intervalNumber intervalUnit]

DO

statements

The schedule is defined in "AT timestamp" (e.g., AT now(), AT '2011-01-01 00:00:00'). You could specify a relative time with the optional "+ INTERVAL", (e.g., AT now() + 1 HOUR).

You need to enable the event scheduler, which is a special thread for maintaining the event queue and running scheduled events:

-- Enable Event Scheduler thread

mysql> **SET @@global.event\_scheduler = ON;**

-- Show the current processing threads

mysql> **SHOW PROCESSLIST \G**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Id: 1

......

Info: SHOW PROCESSLIST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Id: 2

User: **event\_scheduler**

Host: localhost

db: NULL

Command: Daemon

Time: 329

State: Waiting on empty queue

Info: NULL

You can also start the server mysqld with option --event-scheduler=DISABLED|ENABLED.

**Example:**

mysql> **SELECT \* FROM products WHERE productCode LIKE 'PIL%';**

+-------------+-----------+-------------+-----------+--------------+

| productCode | name | description | unitPrice | unitsInStock |

+-------------+-----------+-------------+-----------+--------------+

| PIL-0001 | Pencil 2B | 2B Pencil | 0.54 | 2000 |

+-------------+-----------+-------------+-----------+--------------+

-- Check whether event scheduler is enabled

mysql> **SELECT @@global.event\_scheduler;**

+--------------------------+

| @@global.event\_scheduler |

+--------------------------+

| OFF |

+--------------------------+

-- Enable event scheduler

mysql> **SET @@global.event\_scheduler = ON;**

-- In mysqld console --

100426 21:58:20 [Note] Event Scheduler: scheduler thread started with id 2

-- Schedule an event to increase the price by 10% for some products

mysql> **CREATE EVENT testEvent**

**ON SCHEDULE AT CURRENT\_TIMESTAMP + INTERVAL 30 SECOND**

**DO**

**UPDATE products SET unitPrice = unitPrice \* 1.1 WHERE productCode LIKE 'PIL%';**

mysql> **SHOW EVENTS \G**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Db: southwind\_mini

Name: testEvent

Definer: myuser@localhost

Time zone: SYSTEM

Type: ONE TIME

Execute at: 2010-04-26 21:49:21

Interval value: NULL

Interval field: NULL

Starts: NULL

Ends: NULL

Status: ENABLED

Originator: 0

character\_set\_client: latin1

collation\_connection: latin1\_swedish\_ci

Database Collation: utf8\_unicode\_ci

1 row in set (0.03 sec)

-- 30 seconds later, in mysqld console --

100426 21:59:05 [Note] Event Scheduler: Last execution of southwind\_mini.testEvent. Dropping.

100426 21:59:05 [Note] Event Scheduler: Dropping southwind\_mini.testEvent

100426 21:59:05 [Note] Event Scheduler: [myuser@localhost][southwind\_mini.testEvent]

Data truncated for column 'unitPrice' at row 0

mysql> **SHOW EVENTS \G**

Empty set (0.04 sec)

mysql> **SELECT \* FROM products WHERE productCode LIKE 'PIL%';**

+-------------+-----------+-------------+-----------+--------------+

| productCode | name | description | unitPrice | unitsInStock |

+-------------+-----------+-------------+-----------+--------------+

| PIL-0001 | Pencil 2B | 2B Pencil | 0.59 | 2000 |

+-------------+-----------+-------------+-----------+--------------+

An event can also be recurring:

CREATE EVENT eventName ON SCHEDULE

EVERY intervalNumber intervalUnit

[STARTS startTimestamp [+ INTERVAL count interval]]

[ENDS endTimestamp [+ INTERVAL count interval]]

DO SQLstatements

**Example:**

-- Write an event to backup the Customers table daily

CREATE EVENT backupCustomers

ON SCHEDULE

EVERY 1 DAY

STARTS NOW()

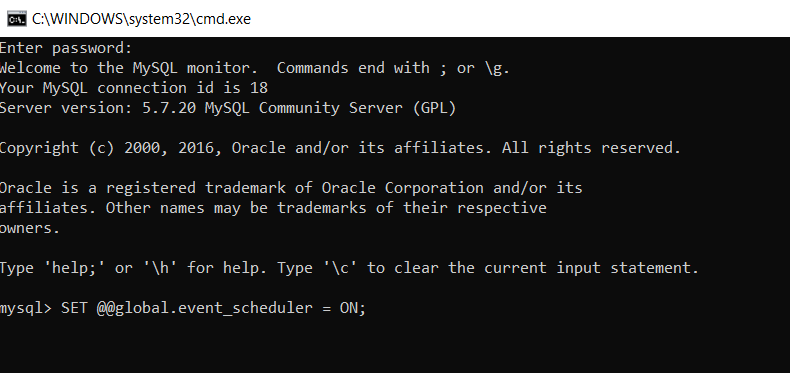
ON COMPLETION PRESERVE

COMMENT 'Daily copy of customers table to backup\_customers'

DO

......

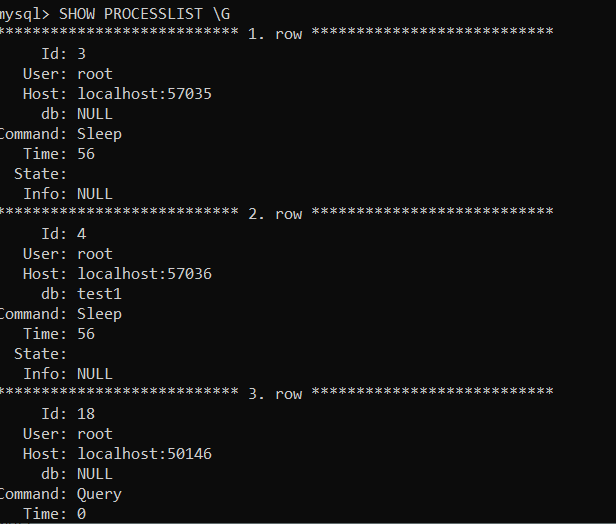
**Step 1**



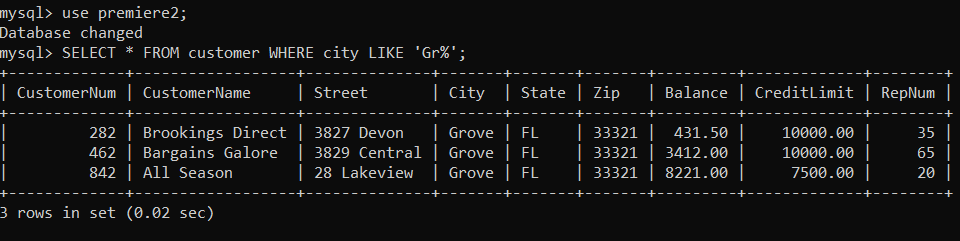
## **MySQL event scheduler configuration**

MySQL uses a special thread called event scheduler thread to execute all scheduled events. You can view the status of the event scheduler thread by executing the SHOW PROCESSLIST command:

**Step 2**



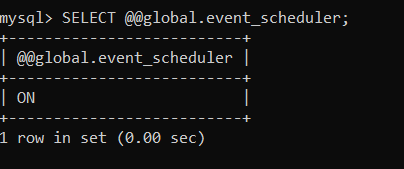
**Step 3**



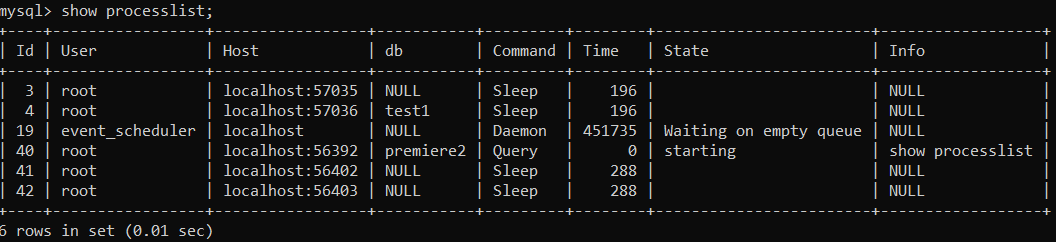
**Step 4**

-- Check whether event scheduler is enabled

If the event scheduler is not enabled, you can set he event\_scheduler system variable to enable and start it:

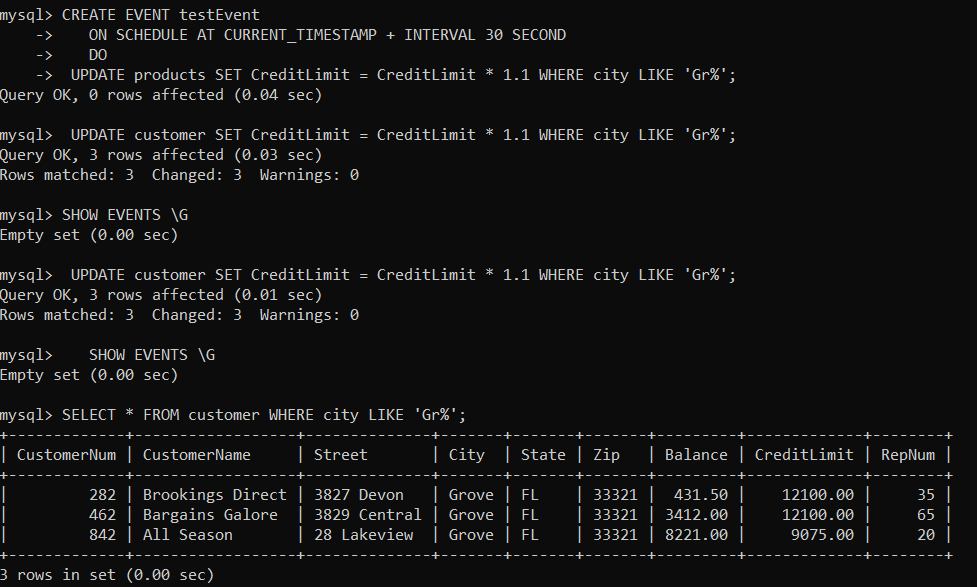


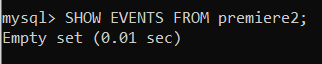
**Check Processlist again**



**Step 5**

-- Schedule an event to increase the CreditLimit by 10% for customers who live in Grove





The output shows no row because the event is automatically dropped when it is expired. In this case, it is a one-time event and expired when its execution completed.

To keep the event after it is expired, you use the ON COMPLETION PRESERVE clause.

Let us create a table called Messages.

Create a test database.

**CREATE** **TABLE** messages (

**id** INT PRIMARY **KEY** AUTO\_INCREMENT,

message VARCHAR(255) **NOT** NULL,

created\_at DATETIME **NOT** NULL

);

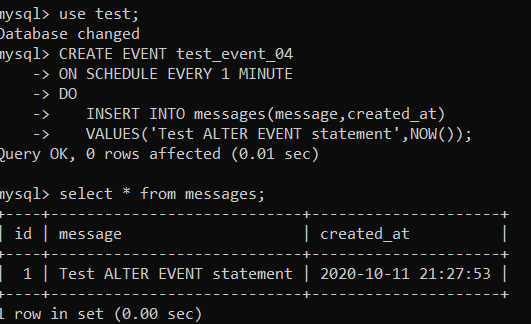
CREATE EVENT test\_event\_04

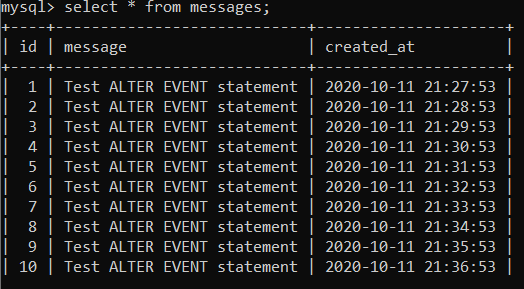
ON SCHEDULE EVERY 1 MINUTE

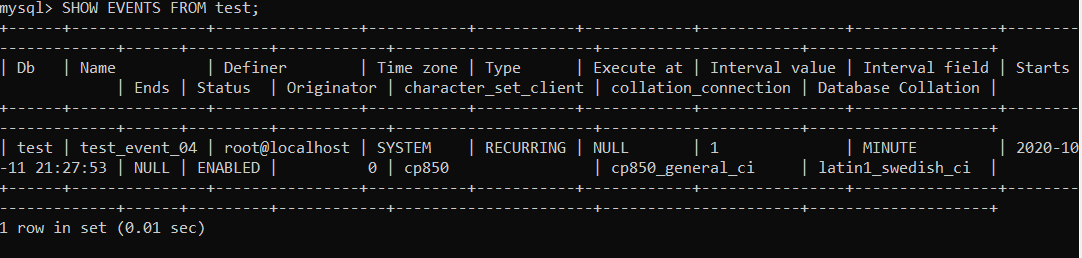
**DO**

INSERT INTO messages(message,created\_at)

VALUES('Test ALTER EVENT statement',NOW());







### **Disable an event**

To disable the event, you use the following statement:

ALTER EVENT test\_event\_04

DISABLE;



ALTER EVENT event\_name

ON SCHEDULE schedule

ON COMPLETION [NOT] PRESERVE

RENAME TO new\_event\_name

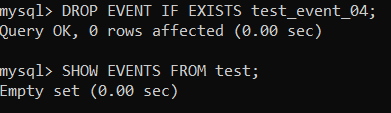
ENABLE | DISABLE

DO

event\_body

**DROP** **EVENT**

DROP EVENT [IF EXISTS] event\_name



### Administering MySQL Database Server

<https://dev.mysql.com/doc/refman/5.7/en/create-user.html>

#### 5.1  Managing User Accounts

To create a new user, use CREATE USER command as follows:

CREATE USER 'username'[@'userhostname'] IDENTIFIED BY PASSWORD 'password';

The default *userHostname* is localhost. You could use either DNS hostname or IP address. Wildcard '%' and '\_' can be used for *userHostname*, e.g., 'peter'@'%' (for all hosts), 'paul'@'\*.abc.com', 'pris'@'128.1.2.%'. For IP address, a netmask can be specified in *IPAddress*/Netmask, e.g., '192.168.1.0/255.255.255.0'.

The new user created has no privileges. You need to grant the appropriate privilege to the user using GRANT command. Privilege can be grant globally (for all tables of all databases), or at the database-level, table-level, column-level. For example,

-- Global privileges (all tables of all databases)

-- Grant all privileges (except the GRANT privilege) on all the databases all the tables (\*.\*)

GRANT ALL ON \*.\* TO 'username'@'userhostname';

-- Grant all privileges, including GRANT privilege

GRANT ALL ON \*.\* TO 'username'@'userhostname' WITH GRANT OPTION;

-- Grant selected privileges on all the databases all the tables

GRANT SELECT, INSERT ON \*.\* TO 'username'@'userhostname';

-- Database-level

-- Grant selected privileges on all the tables of a particular database

GRANT SELECT, INSERT ON databaseName.\* TO 'username'@'userhostname';

-- Table-level

-- Grant selected privileges on selected tables of a particular database

GRANT SELECT, INSERT, UPDATE, DELETE ON databaseName.table1Name, databaseName.table2Name TO 'username'@'userhostname';

-- Column-Level

-- Grant selected privileges on selected columns of tables of a particular database

GRANT SELECT(column1Name, column2Name), INSERT ON databaseName.tableName TO 'username'@'userhostname';

The available privileges are:

* Object Rights: SELECT, INSERT, UPDATE, DELECT, EXECUTE, SHOW VIEW.
* DDL Rights: CREATE, ALTER, DROP, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, REFERENCES, INDEX, FILE.
* Others: ALL, ALL WITH GRANT OPTION, GRANT, CREATE USER, CREATE TEMPORARY TABLES, LOCK TABLES.

To remove privileges, use REVOKE command, e.g.,

-- Remove all privileges

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'username'@'userhostname';

You may need to issue a "FLUSH PRIVILEGES" command to clear and reload temporary caches in MySQL, for the new privileges to take immediate effect.

To remove a user, use DROP USER command as follows:

DROP USER 'username'@'userhostname';

To change the password of a user, use SET PASSWORD command, e.g.,

SET PASSWORD FOR 'username'@'userhostname' = PASSWORD('newPassword');

-- For current user

SET PASSWORD = PASSWORD('newPassword');

To rename a user, use RENAME USER command, e.g.,

RENAME USER 'username'@'userhostname' TO 'newUsername'@'userhostname';

##### Security Considerations on User Management

* The superuser root has no initial password. It is critical to protect superuser root with a strong password.
* Remove the anonymous user (identified by an empty string) ''@'localhost', or ''@'127.0.0.1', which was created during the installation by default.
* All new users must be password protected.
* Never choose a password from dictionary. Use a combination of letters and numbers.
* Do not grant any user (except root) access to mysql.user table (because passwords are stored in this table).
* Do not grant more privileges than necessary. Grant the least privilege necessary to carry out the tasks.
* Never grant privileges to all hosts.
* Run "SHOW GRANTS [FOR 'username'@'hostname']" statement to list the privileges granted to a user (default to current user). For example,
* -- List the privilege for the current user
* mysql> SHOW GRANTS;
* -- List the privilege for another user

mysql> SHOW GRANTS FOR 'username'@'hostname';

You can also run the statement in batch mode with execute (-e) option:

Shell> mysql -u username -p -e "SHOW GRANTS FOR 'selectedUser'@'hostname'"

Alternatively, you can list the privileges (vertically) by querying the mysql.user table.

mysql> SELECT \* FROM mysql.user \G