**JDBC**

**Encryption and Dates**

Create the following table for testing:

create table test2

(password varchar(16),

start\_date date);

To encrypt data using mysql, use

aes\_encrypt(‘data’,’key’);

Where aes is “Advanced Encryption Algorithm” formerly known as the Rijndael algorithm. The data and the key can be any length.

To decrypt, use

aes\_decrypt(‘data’,’key’);

Where the key must be the same used for encrypting.

Insert the following data:

insert into test2

values(aes\_encrypt('hello','key'),now());

Where now() is a function setting the current date.

You can select the data as follows:

select password, aes\_decrypt(password,'key'), start\_date

from test2;

You can insert arbitrary dates using literals as shown below:

insert into test2

values('hello123','2016-10-20 23:12:15');

**JDBC**

* Java programs communicate with databases and manipulate their data using the Java Database Connectivity (JDBC™) *API.*
* A JDBC driver enables Java applications to connect to a database in a particular DBMS and allows you to manipulate that database using the JDBC API.
* JDBC is the same interface for all databases – MySQL, Oracle, etc. Only the low-level driver needs to be changed, but your actual program does not.

Most major Database vendors provide a driver for their database.

MySQL 5.0 Community Edition is an open-source database management system that executes on many platforms, including Windows, Linux, and Mac OS X.

* To use MySQL with JDBC, you also need to install MySQL Connector/J—a JDBC driver that allows programs to use JDBC to interact with MySQL.
* MySQL Connector/J can be downloaded from
  + dev.mysql.com/downloads/connector/j/
* The documentation for Connector/J is located at
  + dev.mysql.com/doc/refman/5.5/en/connector-j.html.

To connect to the MySQL use the following

* An object that implements interface Connection manages the connection between the Java program and the database.

Connection con;

con = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/student\_space","student","hello");

Should use properties files.

* Connection objects enable programs to create SQL statements that manipulate databases.
* DriverManager static method getConnection attempts to connect to the database specified by its URL.
* Three arguments
  + a String that specifies the database URL
  + a String that specifies the username
  + a String that specifies the password
* The jdbc:mysql://localhost:3306/student\_space specifies
  + the protocol for communication (jdbc)
  + the subprotocol for communication (mysql)
  + the location of the database (//localhost:3306/student\_space, where localhost is the host running the MySQL server and student\_space is the database name).
* The subprotocol mysql indicates that the program uses a MySQL-specific subprotocol to connect to the MySQL database.

:

* Connection method createStatement obtains an object that implements interface Statement (package java.sql).
  + Used to submit SQL statements to the database.
* The Statement object’s executeQuery method submits a query to the database.
  + Returns an object that implements interface ResultSet and contains the query results.
  + The ResultSet methods enable the program to manipulate the query result.
* A ResultSet’s ResultSetMetaData describes the ResultSet’s contents.
  + Can be used programatically to obtain information about the ResultSet’s column names and types.
* ResultSetMetaData method getColumnCount retrieves the number of columns in the ResultSet.

Invoke execute on the Statement object in order to perform an SQL DML statement (insert, update, or delete)

You can specify a column number (begin counting at 1) or the column name (less error prone) when accessing the result set:

**while**(rs.next())

{

System.***out***.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3));

}

rs.next() returns a boolean – true if a row exists, false otherwise. Invoking next always fetches the next row from the dbms work area.

SQL exceptions are thrown whenever an SQL syntax is in your executeQuery or execute statement or when trying to access a table that doesn’t exist.