**Activity 05**

**Introduction to SQL and JDBC**

**Please submit your answers to the following questions.**

1. Define the following Key Terms:

* Attribute: a property or characteristic of an entity that we choose to record.
* database management system: it is the system in which related data is stored in an efficient or compact manner.
* database system: a way of organizing information on a computer, implemented by a set of computer programs.
* domain constraint: a user-defined data type which enforces the integrity of the standard data types.
* foreign key constraint: it specifies that the values in a column must match the values appearing in some row of another table.
* integrity constraint: are used to ensure accuracy and consistency of data in a relational database.
* primary key constraint: uniquely identifies each record in a database table.
* relational database: a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.
* Structured Query Language (SQL): a standard interactive and programming language for getting information from and updating a database.
* Tuples: a tuple is one record or one row on a table.

1. What are superkeys, candidate keys, and primary keys? How do you create a table with a primary key?

A superkey is a combination of attributes that can be uniquely used to identify a database record. A candidate key is a field that could be the primary key but was not chosen. A primary key is an attribute or combination of attributes that uniquely identifies a row or record in a relation. A primary key can be created using the command PRIMARY KEY. For example: the column custumer\_id can be made a primary key using PRIMARY KEY(custumer\_id).

1. What is a foreign key? How do you create a table with a foreign key?

A foreign key is a non-key field which is the primary key in another table. A foreign key can be created using the command FOREIGN KEY… REFERENCES. For example: FOREIGN KEY (custumer\_id) REFERENCES Persons(custumer\_id).

1. Can a relation have more than one primary key or foreign key?

A relation can have only one primary key, but can more than one foreign key.

1. Does a foreign key need to be a primary key in the same relation?

No, but when we have a table with a one-to-one relationship, where the foreign key and primary key of the linked table are one and the same.

1. Does a foreign key need to have the same name as its referenced primary key?

No, a foreign key does not need to have the same name as its referenced primary key.

1. Can a foreign key value be null?

Yes, a foreign key value can be null.

1. What are the advantages of developing database applications using Java?

A primary benefit of using the Java programming language is the wide range of packages available for simplifying a variety of programming tasks. One of these tasks is to provide a persistent storage for Java programs. Actually, this can be accomplished using several different techniques, including SQLJ, JDBC.

1. Describe the following JDBC interfaces: Driver, Connection, Statement, and ResultSet.

JDBC drivers implement the interfaces and classes of the JDBC API. Connection interface is used to establish a connection to a database. Statement interface is used to send static SQL statements to the database server and obtain results. ResultSet interface is used to process the results returned from executing an SQL statement.

1. How do you load a JDBC driver? What are the driver classes for MySQL, Access, and Oracle?

There are two ways to load a JDBC driver: using the Class.forName(“sun.jdbc.odbc.JdbcOdbcDriver”) method, or using the java.lang.System property jdbc.drivers setting.

MySQL driver class: com.mysql.jdbc.Driver

Access driver class: sun.jdbc.odbc.JdbcOdbcDriver

Oracle driver class: oracle.jdbc.OracleDriver

1. How do you create a database connection? What are the URLs for MySQL, Access, and Oracle?

In order to create database connection, first we have to load the driver, using Class.forName(DriverName); then get the connection object, Connection con = Driver.getConnection(loaded driver name);

MySQL URL: jdbc:mysql://<host>:<port>/<database\_name>

Access URL: jdbc:odbc:<database\_name>

Oracle URL: jdbc:oracle:thin:@//[HOST][:PORT]/SERVICE

1. How do you create a Statement and execute an SQL statement?

In order to create a statement, first we have to load the driver, using Class.forName(DriverName); then get the connection object, Connection con = Driver.getConnection(loaded driver name); then create a SQL statement , Statement s = con.createStatement(); then to execute the SQL statement we should create Resultset object using the statement created above, ResultSet rs= s.executeQuery("sql statement");

1. How do you retrieve values in a ResultSet?

To retrieve values in a ResultSet, we can use getString method.

1. Does JDBC automatically commit a transaction? How do you set auto-commit to false?

Yes, by default JDBC Connection is in auto-commit mode so every SQL statement is committed to the database upon its completion. To set auto-commit to false, we can use the following SQL command: Connectionobjectname.setAutoCommit(false);

1. Describe prepared statements. How do you create instances of Prepared-Statement? How do you execute a PreparedStatement? How do you set parameter values in a PreparedStatement?

Prepared Statement is an object that represents a precompiled SQL statement.

To create instances of Prepared-Statement: PreparedStatement pStmt = con.prepareStatement(query);

To execute a Prepared Statement we can use PreparedStatement.executeQuery() method.

To set parameter values in a Prepared Statement is be done by calling setXXX() methods.

1. What are the benefits of using prepared statements?

The benefits of using prepared statements are PreparedStatement object are a lot faster and highly efficient than using a Statement object. Prepared Statements are precompiled ones, so they are not compiled everytime whereas Statement object it is not the case.

1. What is DatabaseMetaData for? Describe the methods in DatabaseMetaData. How do you get an instance of DatabaseMetaData?

Database Metadata describes the structural components of tables and their elements. There are three method in Database MetaData:

dbm.supportsANSI92EntryLevelSQL();

dbm.supportsANSI92IntermediateSQL();

dbm.supportsANSI92FullSQL();

To get an instance of Database Metadata: DatabaseMetaData databaseMetaData = connection.getMetaData();

1. What is ResultSetMetaData for? Describe the methods in ResultSetMetaData. How do you get an instance of ResultSetMetaData?

ResultSetMetaData is used to get the information about the ResultSet().Some of the methods of ResultSetMetaData object:

getColumnName() // returns column name

getColumnType() // returns column data type

getColumnCount() // # of columns in the row

getTableName() //returns the name of the table

1. How do you find the number of columns in a result set? How do you find the column names in a result set?

To find the number of columns in a result set we should use ResultSetMetaData object. And then call getColumnCount() method and To find the column names we should call getColumnName() method.