1. Which ResultSet methods are used extract data out of the ResultSet? (Select all that apply.) 1 / 1 point

- [x] getString(int)

Correct
True; retrieve the value of the designated column by index in the current row of this ResultSet object as a String.

- [ ] beforeFirst()
- [ ] absolute(int)
- [x] getString(Column name)

Correct
True; retrieve the value of the designated column by column name in the current row of this ResultSet object as a String.

2. Which ResultSet methods are used to position a cursor at the start of the ResultSet? 1 / 1 point

- ( ) absolute(int)
- ( ) getString(Column name)
- ( ) getString(int)
- (•) beforeFirst()

Correct
True; moves the cursor to the front of this ResultSet object, just before the first row.

3. How can we define an AUTO Increment column in SQL as a primary key? 1 / 1 point

- ( ) ID INT NOT NULL
- ( ) ID INT
- (•) ID int GENERATED ALWAYS AS IDENTITY not null primary key.
- ( ) ID INT primary ley

Correct
True; the id is auto increment courtesy of the IDENTITY Column.

4. How would you define a foreign key relationship between an Address a table with an auto increment primary key and a foreign key on the Column EMPL_ID with an Employee table whose Primary Key is ID? 1 / 1 point

- ( ) CREATE TABLE ADDRESS(ADDR_ID int GENERATED ALWAYS AS IDENTITY not null primary key, NAME CHAR(50) NOT NULL, EMPL_ID INT, FOREIGN KEY (EMPL_ID) REFERENCES EMPLOYEE(EMPL_ID))
- ( ) CREATE TABLE ADDRESS(ADDR_ID int GENERATED ALWAYS AS IDENTITY not null primary key, NAME CHAR(50) NOT NULL, EMPL_ID INT NOT NULL)
- (•) CREATE TABLE ADDRESS(ADDR_ID int GENERATED ALWAYS AS IDENTITY not null primary key, NAME CHAR(50) NOT NULL, EMPL_ID INT NOT NULL, FOREIGN KEY (EMPL_ID) REFERENCES EMPLOYEE(ID))
- ( ) CREATE TABLE ADDRESS(ADDR_ID int not null primary key, NAME CHAR(50) NOT NULL, ID INT NOT NULL, FOREIGN KEY (ID) REFERENCES EMPLOYEE(EMPL_ID))

Correct
True,; the ADDRESS table has an IDENTITY column auto incrementing its primary key, it has the column EMPL_ID defined as a foreign key to the table EMPLOYEE and its primary key ID.

5. What is referential integrity? Select One answer. 1 / 1 point

- ○ Describes the business relationships represented in the database schema. It ensures that relationships between tables remain consistent.
- ○ Frequently leverages Primary and Foreign Key relationships.
- ◉ All the above.
- ○ Ensures a dependent row in a table with a foreign key cannot exist without its parent with the Primary key it is related too.

Correct
True, they are all true.

6. What is a JdbcRowSet? (Select all that apply.) 1 / 1 point

- ☑ Has the setCommand method for the SQL to be executed.

Correct
True; setCommand(SQL) injects the SQL to be executed by the JdbcRowSet.

- ☑ An enhanced ResultSet.

Correct
True; it has all the properties of ResultSet and some of its own.

- ☑ Supports Result sets sensitive to changes in the underlying objects.

Correct
True; since it is essentially a ResultSet we can set ResultSet.TYPE_SCROLL_INSENSITIVE to pick up changes to the underlying objects.

- ☑ Provides a different syntax to connect to a database than getting a Connection from a DriverManager.

Correct
True; using RowSetFactory factory
= RowSetProvider.newFactory(); and JdbcRowSet jdbcRs = factory.createJdbcRowSet(); we have a JdbcResultSet

7. When creating a Statement object via a connecton's createStament method, what argument do we use to make it use an updateable ResultSet? 1 / 1 point

- ○ ResultSet.TYPE_SCROLL_INSENSITIVE
- ○ ResultSet.CONCUR_READ_ONLY
- ◉ ResultSet.CONCUR_UPDATABLE
- ○ ResultSet.TYPE_SCROLL_SENSITIVE

Correct
True; this defines that the a ResultSet can update the underlying objects.

1 / 1 point

```
try {
	Class.forName("org.apache.derby.jdbc.EmbeddedDriver");
	Connection conn = DriverManager.getConnection("jdbc:derby:c:/employee");
	Statement stmt =  conn.createStatement();
	ResultSet rs = stmt.executeQuery("Select COUNT(*) from EMPLOYEE");
	System.out.println(rs.getInt(1));
} catch (ClassNotFoundException | SQLException e) {
	e.printStackTrace();
}
```

8. What is the output when the following code is run with a JDBC driver if the "employee" table in the employee database exists?

- ○ 1
- ○ The code throws an exception at runtime.
- ◉ 0
- ○ The code does not compile.

Correct
True; we have not inserted anything.

9. ResultSet information is retrieved by using the method: 1 / 1 point

- [x] getObject(column index)

Correct
True; although we have methods getString(int), getInt(int), getDouble(int) etc we can always use the getObject method and cast the resultant.

- [x] getObject(column index, String.class)

Correct
True; this casts the result to a String.

- [ ] getStatement()
- [ ] close()

```
try {
	Class.forName("org.apache.derby.jdbc.EmbeddedDriver");
	Connection conn = DriverManager.getConnection("jdbc:derby:c:/employee");
	Statement stmt =  conn.createStatement();
	ResultSet rs = stmt.executeQuery("Select COUNT(LASTNAME) from EMPLOYEE");
	System.out.println(rs.getInt(1));
} catch (ClassNotFoundException | SQLException e) {
	e.printStackTrace();
}
```

1 / 1 point

10. What is the output when the following code is run with a JDBC driver if the "employee" table in the employee database exists wit COLUNNS ID, FIRSTNAME?

- (•) The code throws an SQLSyntaxException subclass of SQLException at runtime.
- ( ) 1
- ( ) 0
- ( ) The code throws a ClassNotFoundException.

Correct
True, because of the invalid SQL as there is no COLUMN LASTNAME in the table.