



ג'אווה מתקדם JDBC

הרצאה – נחום ליטבין

המצגת והקוד לדוגמה



➤ הקוד והמצגת נמצאים ב GIT
REPOSITORY הבא:

➤ [https://github.com/NahumLitvin/
JDBCExample](https://github.com/NahumLitvin/JDBCExample)

➤ כל COMMIT אמור להוות שלב לוגי
בהרצאה

נושאי ההרצאה

- עקרונות בסיסי נתונים יחסיים ושפת SQL לאחזור ועדכון מידע.
- התקנות
- טעינת דרייברים,
- חיבור לבסיס הנתונים,
- ביצוע שאילתות
- Statement)
- Prepared Statement+Sql Injection
- Called Statment
- עבודה עם ResultSet Rowset
- ביצוע טרנזאקציות

בסיס נתונים יחסי

➤ הסטנדרט דה-פאקטו של בסיסי נתונים עד
2009

➤ בימינו יש גם בסיסי נתונים מסוגים אחרים
כמו Neo4J ו MongoDB

Tables - טבלאות ➡
Keys - מפתחות ➡
Relations - קשרים ➡



מבנה טבלה

שדה - Field ➡

➡ פריט מידע בודד: ת.ז., שם, ת. לידה...

➡ מיוצג ע"י עמודה בטבלה.

רשומה - Record ➡

➡ ישות.

➡ אוסף שדות – רלוונטיים לישות במערכת המידע

➡ מיוצג ע"י שורה בטבלה.

■ מפתח ראשי – Primary Key

■ שדה או קבוצת שדות המזהים רשומה באופן חד ערכי.

■ מפתח זר – Foreign Key

■ שדה בטבלה המקושר לשדה בטבלה אחרת בה משמש השדה כמפתח ראשי.

סוגי קשרים

➤ מטרת הקשרים לדאוג לשלמות הנתונים – Data Integrity.

➤ סוגי קשרים

1:1 יחיד ליחיד – One to One ➤

1:N יחיד לרבים – One to Many ➤

N:M רבים לרבים – Many to Many ➤



על שפת SQL

Structured Query Language ➡

פקודות עיקריות: ➡

Insert ➡ - הוספת רשומה לטבלה.

Select ➡ - שליפת נתונים מטבלה/טבלאות.

Update ➡ - שינוי נתונים בטבלה.

Delete ➡ - מחיקת נתונים מטבלה .

פקודות SQL שונות

- **SELECT** *column_name,column_name*
FROM *table_name*;
- **SELECT** *column_name,column_name*
FROM *table_name*
WHERE *column_name operator value*;
- **UPDATE** *table_name*
SET *column1=value1,column2=value2,...*
WHERE *some_column=some_value*;
- **SELECT** *Orders.OrderID, Customers.CustomerName, Orders.OrderDate*
FROM *Orders*
INNER JOIN *Customers*
ON *Orders.CustomerID=Customers.CustomerID*;

מה זה JDBC

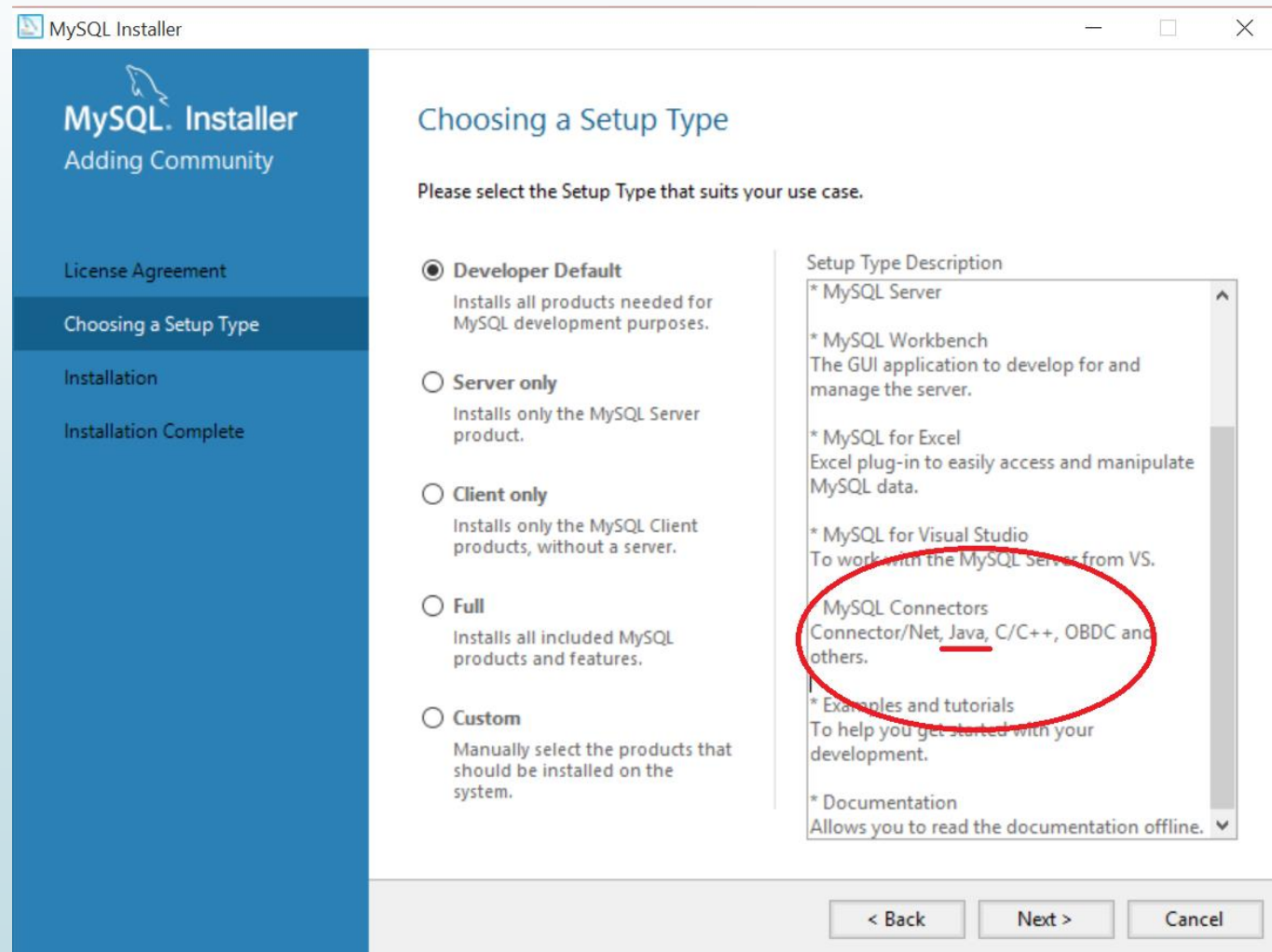
- כל Database עובד קצת אחרת.
- כלומר לא ניתן לצפות כי MySQL יעבוד עם אותו דרייבר של ג'אווה כמו JavaDB
- JDBC מספק ממשק אחוד עבור במתכנתי ג'אווה
- חברה שרוצה שה Database שלה יתמוך ב ג'אווה צריכה ליצור עבורו Jdbc Driver



התקנת MySQL

- MySQL זה אחד מה Databases הפופלאריים בעולם
- אפשר להוריד את MySQL מהאתר הרשמי שלהם מהכתובת
<https://dev.mysql.com/downloads/installer/>
- ההתקנה של MySQL מתקינה לבד גם את ה Driver של JDBC
- שימו לב ל בהתקנה מותקן Connector/J

התקנת JDBC





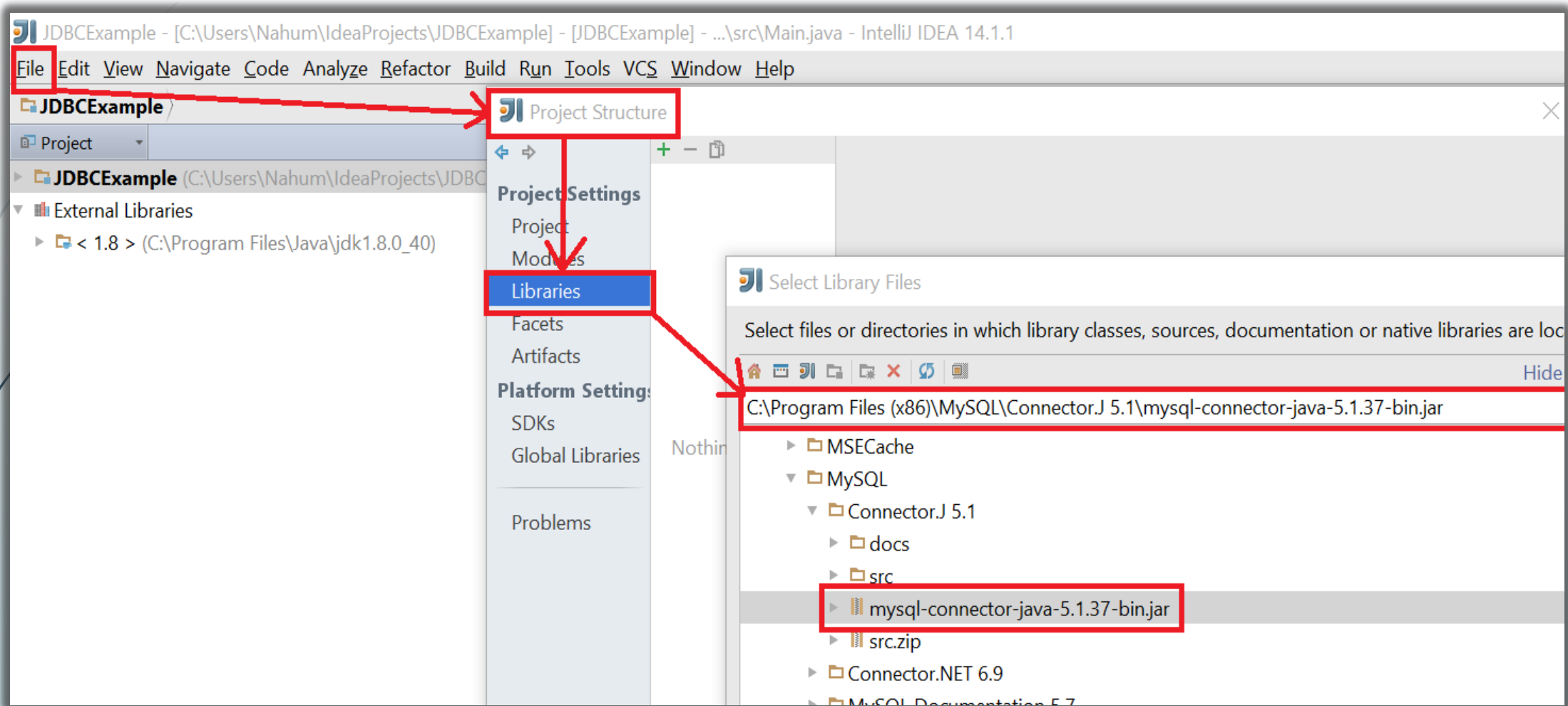
IntelliJ IDEA

IDEA זו סיבת עבודה פופולארית מאד ל JAVA ►

בין השאר ANDROID STUDIO מבוסס עליה. ►

<http://download.jetbrains.com/idea/ideaIC-15.0.1.exe> ►

הוספת Driver לסביבה



התחברות לשרת

Probably the path to Git executable is not valid. [Fix it](#)

Statement ResultSet Select

```
9  Connection conn=null;Statement statement = null;ResultSet resultSet = null;
10 try {
11     conn = getConnection();
12     statement = conn.createStatement();
13     final String SQLQuery = "SELECT * FROM world.city";
14     resultSet = statement.executeQuery(SQLQuery);
15     while (resultSet.next()) {
16         int Id = resultSet.getInt("ID");
17         String Name = resultSet.getString("Name");
18         String CountryCode = resultSet.getString("CountryCode");
19         String District = resultSet.getString("District");
20         int Population = resultSet.getInt("Population");
21         System.out.println("Id:" + Id + " Name:" + Name + " Code"+ CountryCode);
22     finally {
23         if(resultSet!=null)resultSet.close();
24         if(statement!=null)statement.close();
25         if(conn!=null)conn.close();
    }
```

Updatable ResultSet

```
private static void ScrollableAndUpdatable(Connection
conn) throws SQLException{
    //    allow moving forward/backward/edit
    Statement statement = conn.createStatement(
        ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_UPDATABLE; (
    String sql = ;"ytic.dlrow * TCELES"
    ResultSet resultSet = statement.executeQuery(sql);
    resultSet.moveToInsertRow();()
    resultSet.updateString(;"ytiC" , "emaN"
    resultSet.updateString(;"ESP" , "edoCyrtnuoC"
    resultSet.updateString(;"norbeH" , "tcirtsid"
    resultSet.updateInt(;"40452532" , "noitalupoP"
    resultSet.insertRow();()
    {
```

Insert Into – Execute.Update

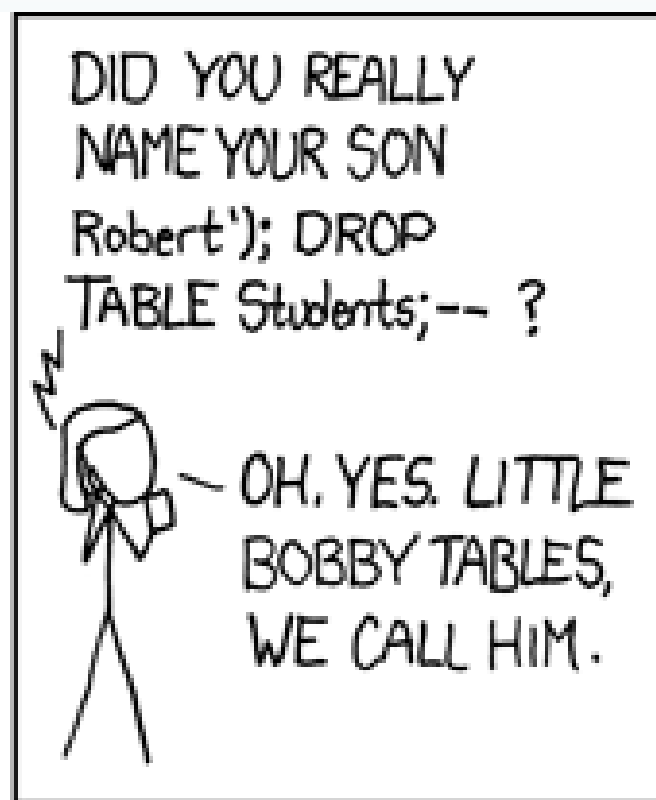
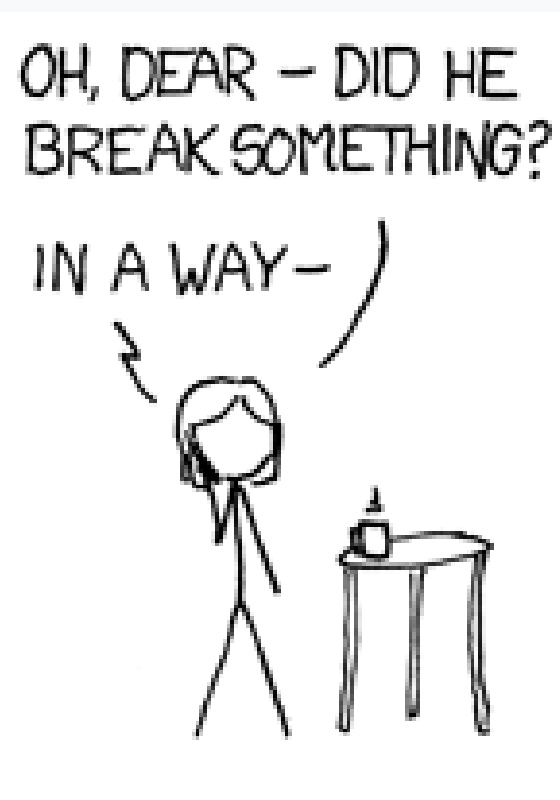
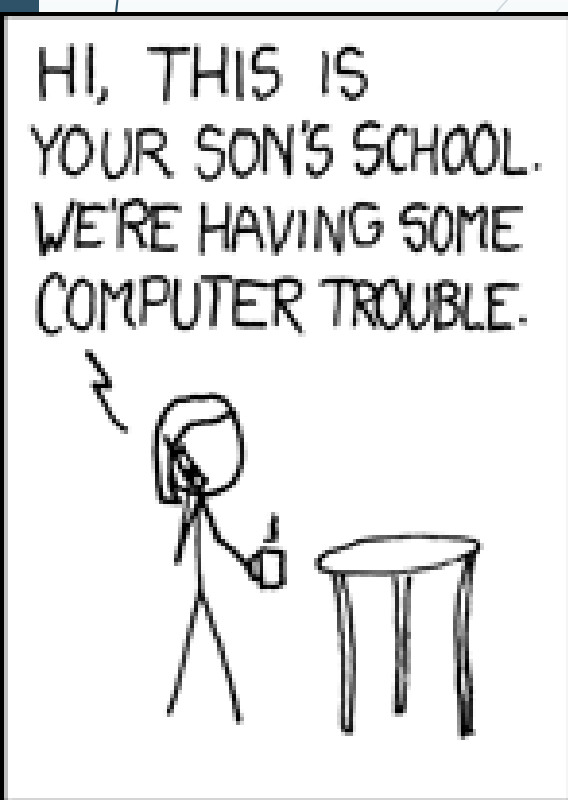
```
20 private static void AddAnewRowToWordCountryTable(Connection conn) throws SQLException
21     Statement statement = null;
22     int linesUpdated;
23     try {
24         statement = conn.createStatement();
25         final String SQLQuery = "insert into world.city"
26             + " (Name, CountryCode, District, Population) "
27             + " values ('MyNewCountry', 'PSE', 'Rafah', 99999) ";
28         linesUpdated = statement.executeUpdate(SQLQuery);
29         if (linesUpdated == 0) { System.out.println("error"); }
30     } finally {
31         if (statement != null) statement.close();
32     }
33 }
```

הזרקת SQL – SQL INJECTION

- שיטה להזריק קוד זדוני לתוך המערכת
- ניצול חוסר בדיקה של משתני קלט

```
Statement statement = conn.createStatement();
String myPassword="Secret";
statement.execute("SELECT * FROM users WHERE password='"+ myPassword+ "'");
//SELECT * FROM users WHERE password='Secret'
myPassword = "' OR 'a'='a";
statement.execute("SELECT * FROM users WHERE password="+ myPassword + "'");
//SELECT * FROM users WHERE password='' OR 'a'='a'
```

וכעת לאתנחתא קומית



Prepared Statement

- שיטה מובנת להגנה מקוד זדוני
- השאילתא נשמרת בצורה מקומפלת.
- שימו לב הספירה מתחילה מ 1

```
myPassword = "' OR 'a'='a";  
PreparedStatement preparedStatement = conn.prepareStatement(  
    "SELECT * FROM users WHERE password = ?");  
preparedStatement.setString(1, myPassword);  
//Throws Exception
```

Called Statement

■ שיטה לקורא ל פרוצדורה מאוחסנת (באנגלית: **Stored Procedures**) (

■ זה בעצם אוסף שאלות SQL שמאוחסן בתוך מסד הנתונים מה שמקטין את סיבוביות הממשק של המפתח ומשפר ביצועים.

```
CallableStatement cs = conn.prepareCall("{call SHOW_SUPPLIERS  
ResultSet rs = cs.executeQuery();
```

RowSet

RowSet אמור לפשט עבודה מול JDBC ולהרחיב את היכולות לשל
ResultSet

יש המון סוגי סטים
JdbcRowSet, - מחובר – מאפשר גלילה ו עדכון גם אם מסד הנתונים לא

תומך

FilteredRowSet,JoinRowSet, WebRowSet, CachedRowSet
לא מחוברים – ניתנים לסרליזציה\ פשוטים יותר – ניתן להעביר דרך האינטרנט

כולם מתפקדים בתור רכיבים של JavaBeans

כלומר יש להם Properties שניתנים לעריכה ב ממשק המשתמש

יש להם תמיכה במערכת האירועים של JavaBeans

שימוש ב ROWSET

```
private static void JdbcRowSetDemo() {  
    try {  
        Class.forName("com.mysql.jdbc.Driver");  
        JdbcRowSetImpl rowSet = new JdbcRowSetImpl();  
        rowSet.setUrl("jdbc:mysql://localhost");  
        rowSet.setUsername("root");  
        rowSet.setPassword("root");  
        rowSet.setCommand("SELECT * FROM emp");  
        rowSet.execute();  
    } catch (ClassNotFoundException | SQLException ex) {  
        ex.printStackTrace();  
    }  
}
```

טרנסאקציות - הסבר

- הדרך שלנו לסנכרן מספר פעולות במסד נתונים
- לדוגמה הוצאת כסף מחשבון בנק אחד והעברה לאחר
- אנחנו רוצים לוודא שאף אחת מהפעולות לא נכשלה
- ואם משהו נכשל לבטל את הכל

טרנסאקציות דוגמה

- ממש כמו קוד TRY CATCH FINALLY רק למסדי נתונים

```
private static void Transaction(Connection conn) throws SQLException{
    try {
        conn.setAutoCommit(false);
        //DO SQL STUFF
        conn.commit();
    }
    catch (Exception e)
    {
        conn.rollback();
    }
}
```

הפניות

שימוש ב-JDBC Driver <https://dev.mysql.com/doc/connector-j/en/> ➤

התקנת Mysql <https://dev.mysql.com/downloads/installer/> ➤

התקנת IDEA <http://download.jetbrains.com/idea/ideaIC-15.0.1.exe> ➤

הזרקת SQL <https://he.wikipedia.org/wiki/> ➤

<https://xkcd.com/327/> ➤

<https://docs.oracle.com/javase/tutorial/jdbc/> ➤

<https://docs.oracle.com/javase/tutorial/jdbc/basics/rowset.html> ➤

שאלות??

