

# JDBC

## Java Database Connectivity



# مقدمه

- Database
  - Collection of data
- DBMS
  - Database management system
  - Stores and organizes data
- SQL
  - Relational database
  - Structured Query Language

# Relational-Database Model

Row	Number	Name	Department	Salary	Location
	23603	Jones	413	1100	New Jersey
	24568	Kerwin	413	2000	New Jersey
	34589	Larson	642	1800	Los Angeles
	35761	Myers	611	1400	Orlando
	47132	Neumann	413	9000	New Jersey
	78321	Stephens	611	8500	Orlando
Primary key			Column		

جدول Employee

# SQL

SQL keyword	Description
SELECT	واکشی از یک یا چند جدول
FROM	جداولی که در لیست SELECT قرار است استفاده شوند
WHERE	شرایطی که اطلاعات واکشی شده را فیلتر کند . همچنین برای به روز رسانی و حذف اطلاعات
GROUP BY	گروه بندی اطلاعات
ORDER BY	مرتب سازی اطلاعات
INNER JOIN	ادغام سطر ها از چند جدول
INSERT	ایجاد سطر در جدولی خاص
UPDATE	به روز رسانی اطلاعات یک یا چند سطر در جدولی خاص
DELETE	حذف یک یا چند سطر از جدولی خاص

# ODBC ( Open Database Connectivity)

ODBC تکنولوژی است که توسط مایکروسافت ارائه شده و به شما امکان تعریف اتصال به پایگاه های داده ی مختلف را میدهد.

هدف ODBC مستقل کردن اتصالات به سیستم عامل میباشد

# JDBC

- JDBC یک تکنولوژی در استاندارد جاوا است که شامل یک API برای تعریف نحوه ی اتصال به پایگاه های داده میباشد .  
این API شامل موارد زیر میباشد:
- اتصال به پایگاه داده
  - واکنشی اطلاعات
  - به روز رسانی اطلاعات
  - صدا زدن Function ها و Procedure ها
  - استفاده از ODBC برای اتصال

# مراحل اتصال به پایگاه داده با JAVA

(1) بارگذاری درایور

(2) ایجاد یک اتصال (Connection)

(3) مکالمه

(1) ارسال دستور واکنشی اطلاعات و بازیابی اطلاعات

(2) ارسال دستورات تغییر (ایجاد، حذف، به روز رسانی)

(3) فراخوانی دستورات و خواندن خروجی آنها

# بار گذاری درایور

**Class.forName(DRIVER\_NAME)**

Driver class	Start of dbURL	Database
sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:	Bridge to Microsoft ODBC (included with JDK)
jdbc.idbDriver	jdbc:idb:	Instant Database (IDB)
oracle.jdbc.Driver.OracleDriver	jdbc:oracle:thin: @server:port#:dbname	Oracle
postgresql.Driver	jdbc:postgres:// host/database	PostgreSQL (freeware database; see <a href="http://www.postgresql.org">http://www.postgresql.org</a> )
com.mysql.jdbc.Driver	jdbc:mysql://host/ database	MySQL (freeware database; see <a href="http://www.mysql.com">http:// www.mysql.com</a> )



# ایجاد یک اتصال (Connection)

**DriverManager**.getConnection(**DB\_URL** , **USER\_NAME** ,  
**PASSWORD**)

**Class**.forName("oracle.jdbc.driver.OracleDriver");

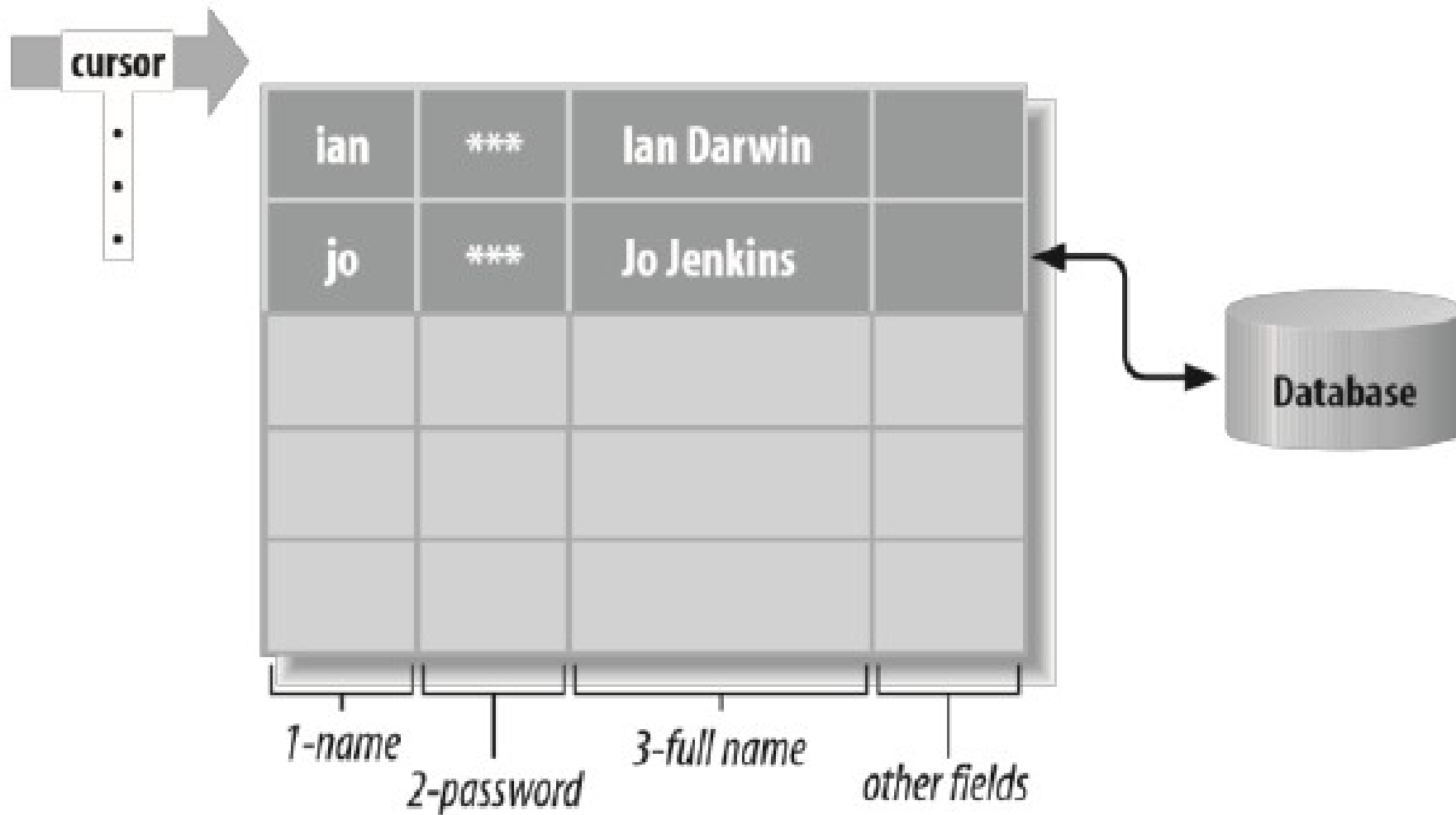
**String** url =

"jdbc:oracle:thin:@//192.168.100.20:1521:XE";

**Connection** conn =

**DriverManager**.getConnection(url,"scott","tiger");

# مکالمه (واکشی)



# مکالمہ (واکشی)

```
Statement stmt = conn.createStatement();
```

```
ResultSet rset =
```

```
    stmt.executeQuery("select * from Employees");
```

```
//For each Rows
```

```
while (rset.next()) {
```

```
    //Print the first column
```

```
    System.out.println (rset.getString(1));
```

```
}
```

```
stmt.close();
```

# مکالمه (واکشی)

JDBC method	SQL type	Java type
getBit()	BIT	boolean
getByte()	TINYINT	byte
getShort()	SMALLINT	short
getInt()	INTEGER	int
getLong()	BIGINT	long
getFloat()	REAL	float
getDouble()	DOUBLE	double
getString()	CHAR	String
getString()	VARCHAR	String
getString()	LONGVARCHAR	String
getDate()	DATE	java.sql.Date
getTimeStamp()	TIME	java.sql.Date
getObject()	BLOB	Object

## مکالمه (DML)

```
Statement stmt = conn.createStatement();
```

```
String sql = "UPDATE Registration SET age = 30 WHERE  
id in (100, 101)";
```

```
stmt.executeUpdate(sql);
```

# استفاده از PreparedStatement ها به جای

## Statement

- مشکل: هر SQL یک رشته ای است که قبل از ارسال باید ترجمه و کامپایل شود. بعضی از SQL ها شبیه به هم هستند تنها مقادیر آنها تغییر میکند:

```
select * from payroll where personnelNo = 12345;
```

```
select * from payroll where personnelNo = 23740;
```

```
select * from payroll where personnelNo = 97120;
```

# استفاده از PreparedStatement ها به جای

## Statement

- راه حل: از کلاس PreparedStatement استفاده کنید ، بخش هایی که متغیر دارند را با «؟» مشخص کنید ، سپس قبل از ارسال SQL مقادیر آنها را مشخص نمایید

```
select * from payroll where personnelNo = ?;
```

```
PreparedStatement pstmt = null;
String sql = "INSERT INTO PERSON (ID,FIRST_NAME,LAST_NAME,STAT_CD) VALUES (?, ?, ?, ?)";
try {
    pstmt = connection.prepareStatement(sql);
    pstmt.setInt(1, 87);
    pstmt.setString(2, "Lokesh");
    pstmt.setString(3, "Gupta");
    pstmt.setInt(4, 5);
    int affectedRows = pstmt.executeUpdate();
    System.out.println(affectedRows + " row(s) affected !!");
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try { pstmt.close();
    } catch (Exception e) {
    }
}
```