JDBC

1. What is the name of the only concrete class in the JDBC API?

Only concrete class: DriverManger

Used to manage the driver

Vendor specific driver

Register driver

1. List the main JDBC API interfaces.

Statement

Resultset

DatabaseMetaData、

1. What is JDBC

JDBC represents for Java Database connectivity. It’s an API that enables Java application to interact with the databases.

1. How to setup: database vendor/url/username and password
2. Some interface:

Connection

* Used to establish the connection
* The link between java application and the database
* Connection is provided by drivermanager
* Vendor specific

Statement

* Used as encapsulating layer for sql statement
* Concretion is provided by Connection and is also vendor-specific

Resultset

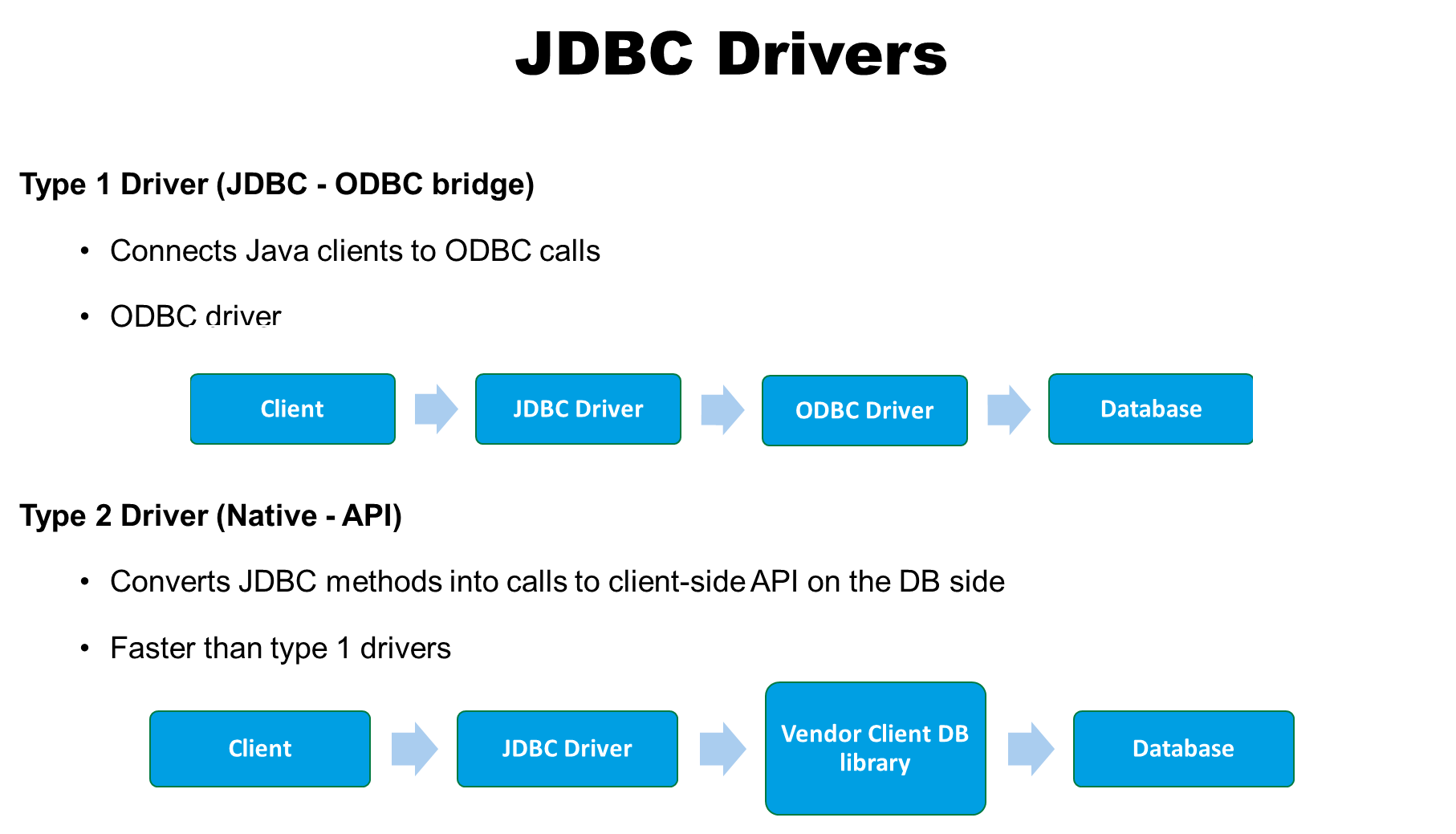
* Encapsulate data returned from the database
* Represents a table of data
* Return when we execute a select query through JDBC

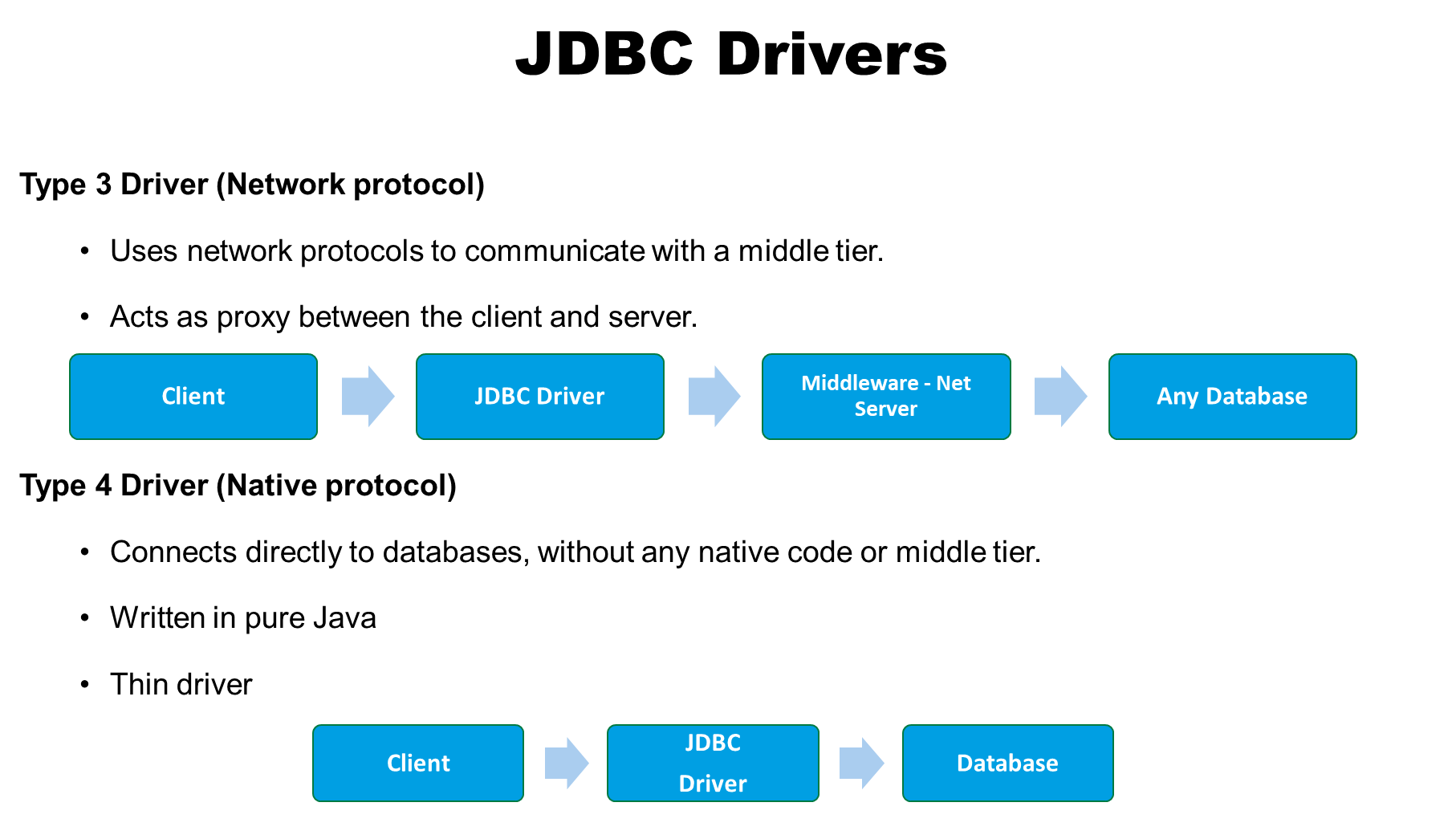
Databasemetadata

* Provides information about the database as a whole

1. What types of driver

JDBC drivers implement the defined interfaces in the JDBC API, for interacting with your database server.





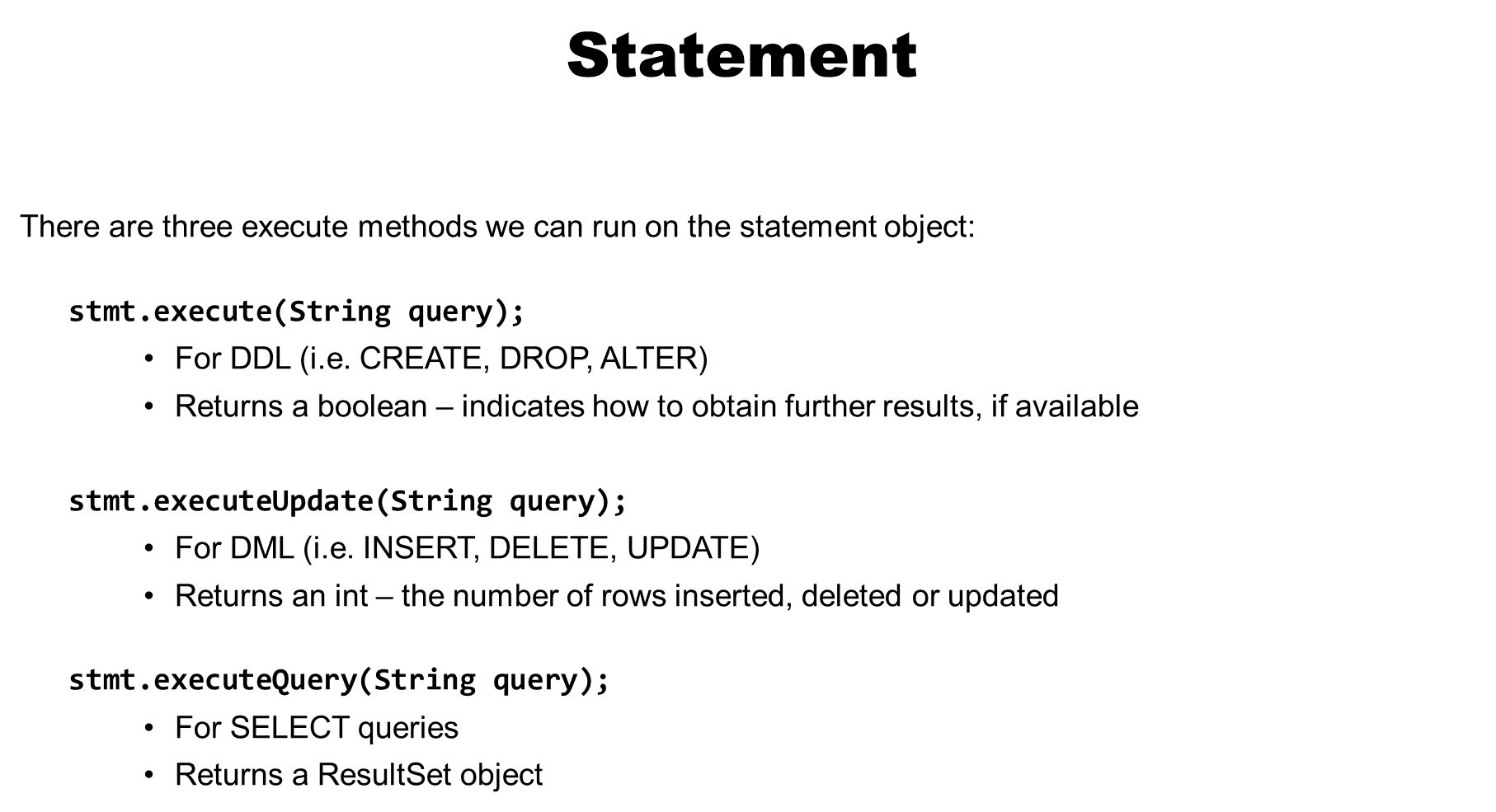
1. How to get connection object?

Drivermanager.getConnection

url: vendor driver type location port ssid: Service Set Identifier

1. Different statement

Statement=connect.createStatement()



1. Why we need next()

The resultset is accessed using a cursor. We need the next() to move the cursor through the result set.

The next return true if there is a row of data.

Return false if it’s an empty row.

1. How we can get data from the ResultSet

getInt()

getString()

1. How we can handling the NULL

Using BigDecimal

Rs.wasNull()

1. Different types of ResultSet

You can define it in the createStatement function: type concurrency, resultsethodability

TYPE\_FORWARD\_ONLY

TYPE\_SCROLL\_INSENSITIVE

TYPE\_SCROLL\_SENSITIVE

Cocurrency

CONCUR\_READ\_ONLY

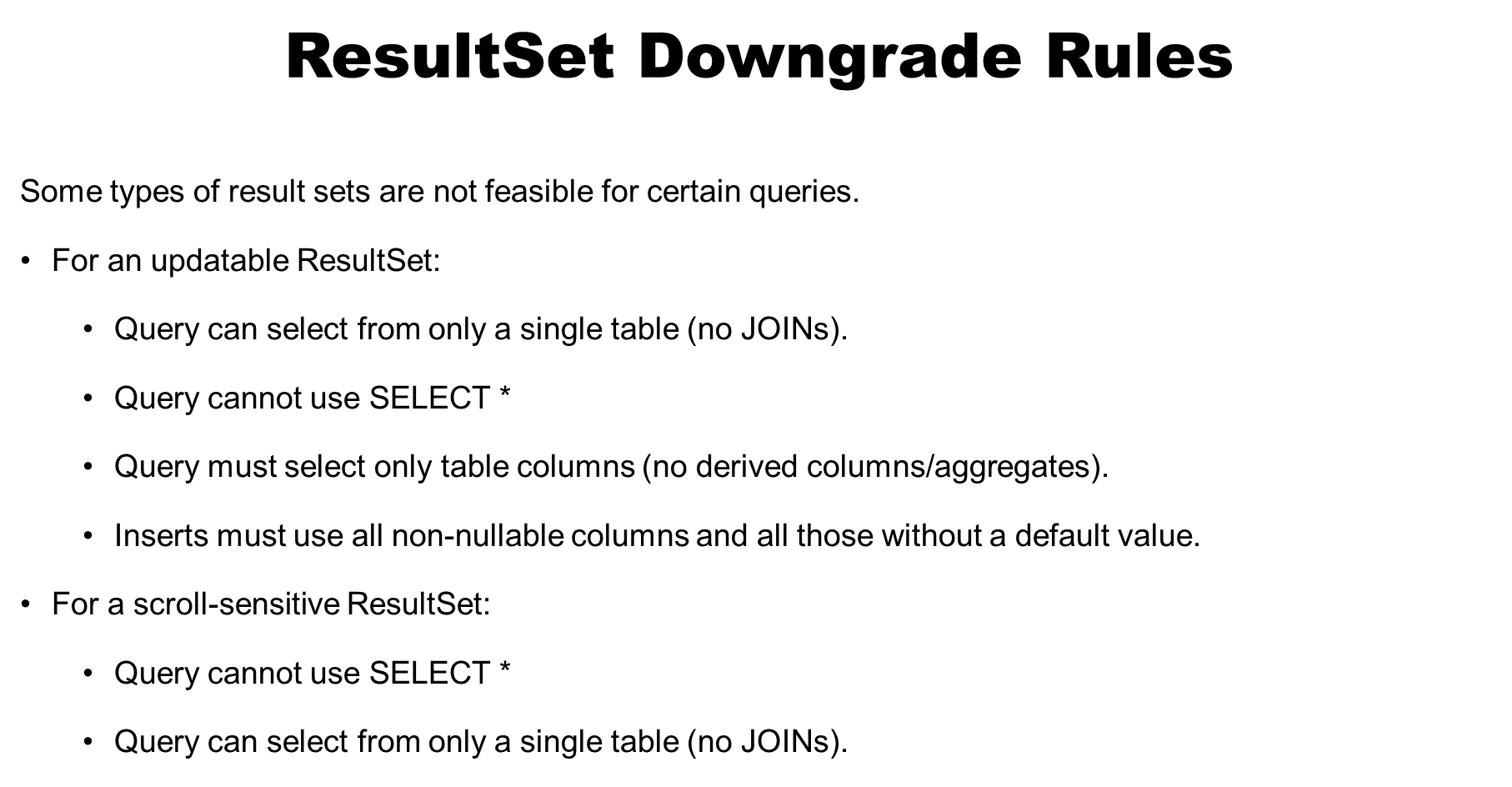
CONCUR\_UPDATABLE

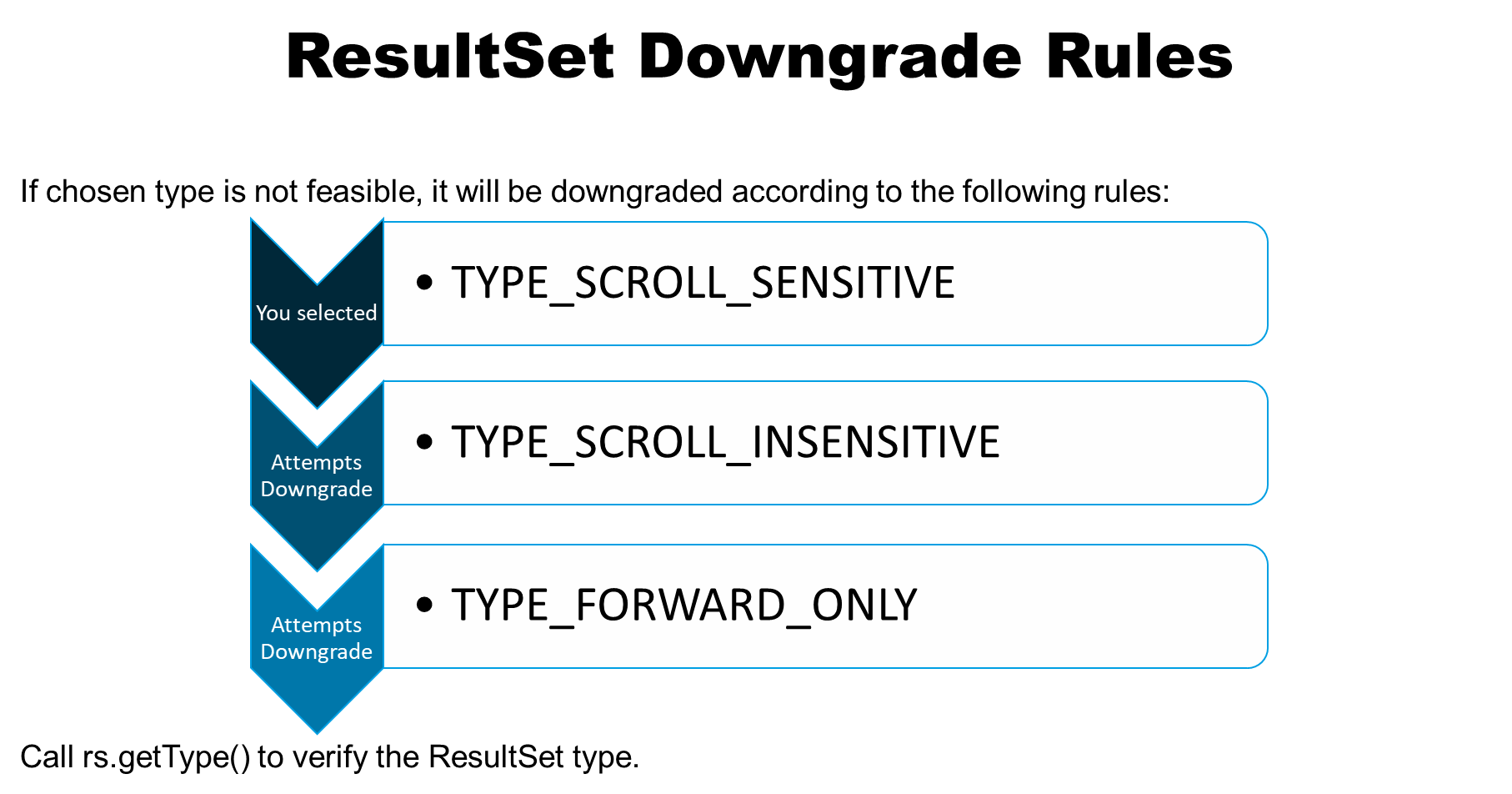
Holdability

HOLD\_CURSORS\_OVER\_COMMIT

CLOSE\_CURSORS\_OVER\_COMMIT

Oracle only support HOLD\_CURSORS\_OVER\_COMMIT





1. Why we use preparedStatement?
2. Prevent the sql injection attacks
3. Reusability improve the performance of executed statements.
4. What is Batching and Why we use it?

Batching is grouping SQL statements together

Decreases the number of calls to the database to 1 – instead of 1 for every statement

SetAutocommit(false)

Conn.commit()

1. Callable Statement

CallableStatement extends PreparedStatement  
It is used for calling stored procedures in the database.

Stored procedures are created for intensive database processes.

This reduces network bandwidth consumption by sending less data over the network.

JPA

1. What is persistence?

Persistence means save the data past the life of the application process using the database to do this.

1. What is ORM

ORM refers to Object relational mapping. It’s a technique maps object state to the data in a relational database.

1. What is JPA

Java persistence API. A ORM framework handle the task of mapping objects to tables.

The ORM solution in Java EE.

1. What is the difference between Hibernate and JPA

It has collected the best ideas from existing persistence technologies like Hibernate, TopLink, and JDO. The result is a standardized specification that helps you build a persistence layer that is independent of any particular persistence provider.

1. Entities

Entity classes are the data we want to persist.

Any object can be an entity:

Persistent

Has an identity

Transactional—transaction needed for changes to DB

Fine-grained

Metadata relates entities to records in the database

Annotations(@Entity)

1. Key interfaces

EntityManagerFactory

EntityManager

EntityTransaction: group operations on persistent data into units

Query : finds persistent objects that meet certain criteira

Persistence

1. Key steps
2. Configure Entity classes with metadata
3. Provide configuration details in persistence.xml file
4. What is detached and managed

EM will automatically sync Entity state with the DB

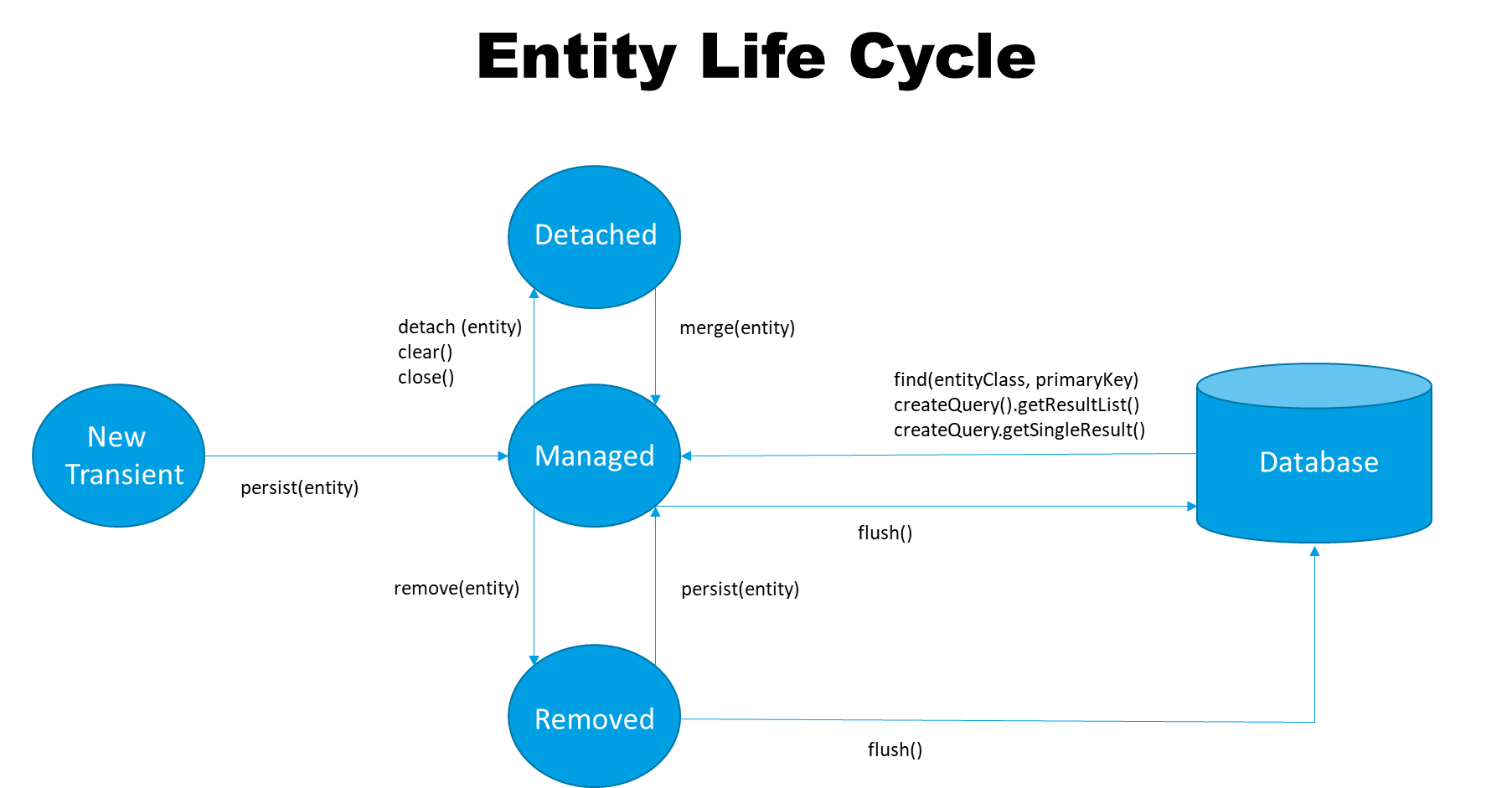
1. What is persistence.xml

**persistence.xml**

* + An XML file that defines **persistence units**
  + Should be placed in a directory called META-INF

A **persistence unit** is used to configure**:**

* + Which Entity classes are to be managed
  + Persistence provider details
  + Database connection details and other properties



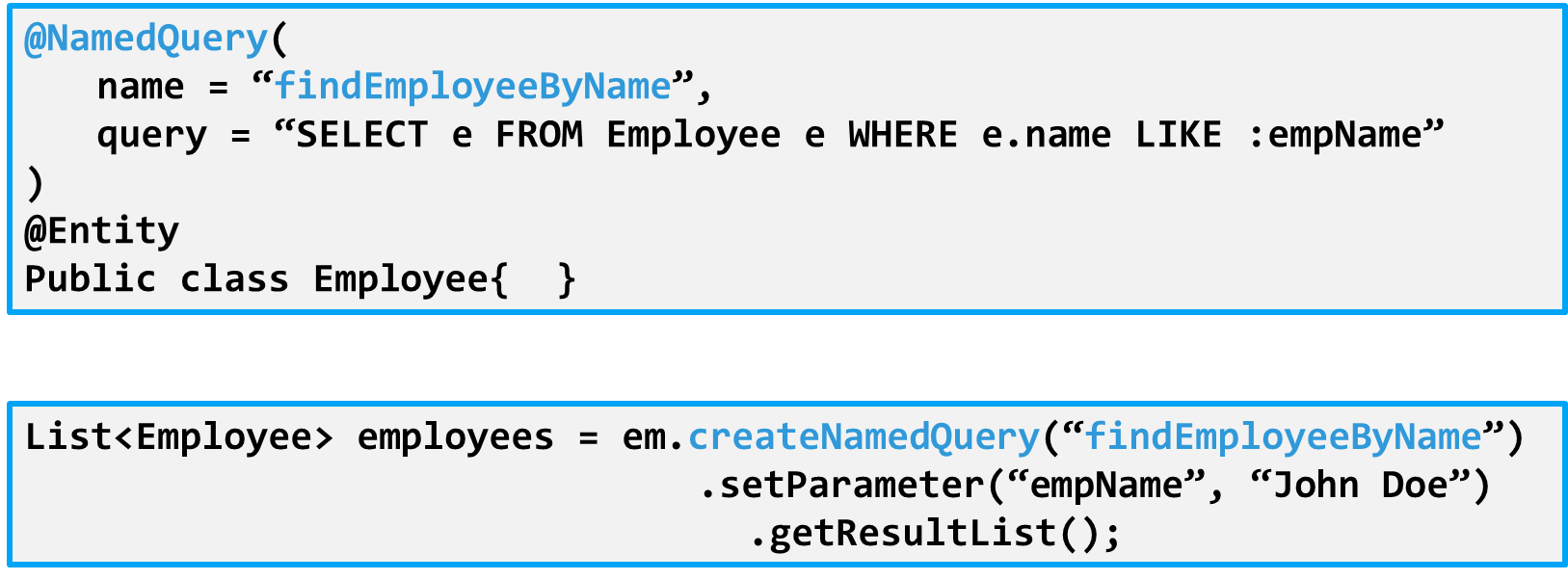
1. What EntityManager do?

Manage Entities and connect the entity with the database.

1. What are the different methods in JPQL to create query?

CreateQuery

CreateNamedQuery



createNativeQuery

createStoredProcedureQuery

1. What is static and dynamic query?

Static query: defined in the metatdata using annotation

Dynamic query: Defined directly within an application’s business logic

1. How do you implement named and position parameter?

:name

?1

1. What are the different primary key generation strategies ?

GenerationType.AUTO

IDENTITY

SEQUENCE

TABLE

1. What are the annotations used for user defined primary key generation ?

@GeneratedValue(strategy,generatorName)

1. What is Entity Relationship

Relationship between two entity classes

1. What are the two factors that defines a relationship.

Direction: whether a query can navigate from one entity to another

Multiplicity: how many objects participate in a relationship

1. What is unidirectional mapping and how to implement it?

Has only one owning side

1. What is bidirectional mapping and how to implement it ?

MappedBy

1. What does cascade operation mean?

Entity in relationship have dependencies on the existence of the other entity in the relationship

1. What is one-to-one mapping and how to implement it?

@One-to-One

1. What is many-to-one and one-to-many mapping and how to implement it?

@Many-to-one

@One-to-Many

List

1. What is many-to-many mapping and how to implement it?

@ManyToMany

@JoinTable(name=“EMPLOYEE\_SKILL”,

joinColumns=

@JoinColumn(name=“FK\_EMP\_ID”), inverseJoinColumns=

@JoinColumn(name=“FK\_SKILL\_ID”)

)

1. What is Entity Inheritance ?

Entity class can extend classes.

1. What are the different types of inheritance strategies ?
   1. Mapped Superclass
   2. Single table per class hierarchy
   3. Table per concrete entity class
   4. Joined Subclass
2. How many tables are created for Joined Subclass strategy ?

3

1. What is the annotation used for creating single table for entire class hierarchy ?

InheritanceType.SINGLE\_TABLE

1. What is the use of @DiscriminatorColumn

The type of the column to be used as a discriminator column. The default is DiscriminatorType.STRING. This element is optional.

1. What is a web container?

Used for generate dynamic webpage

1. What is a Servlet?

Server side java components.

Process HTTP requests and generate Http Response.

1. What is the web container responsible for?

Running servlet code to generate response.

Security

Concurrency

Lifecycle of all components(Filter,JSPs,servlets,listeners)

Application configuration and deployment.

1. What are the lifecycle steps of a Servlet?

Load the class

Create an instance(only one for each servlet)

Initialize by calling init()

Call the service() method

Call the destroy method

1. What is the deployment descriptor?

A configuration file called the **deployment descriptor (web.xml)** specifies how the application is to be deployed:

Servlet

Url mapping

Filter listener

Login configuration

Security constraints

We also use annotation to do this job.

1. List some tags used in the deployment descriptor.

<servlet></servlet>

<servlet-mapping></servlet-mapping>

<url-pattern></url-pattern>

1. Servlet Context

An object that represents the context of the entire application

* Create at startup
* Only one per application
* Contains configuration for the app
* Application scope

1. Session

An httpSession object may be created to maintain a session for a particular client.

getSession

isNew

invalidate

1. Filter benefits

Add functionality without modifying existing Servlets

Add same piece of functionality to multiple servlets

Enhance entire application

doFilter()

1. What is a Listener?

An implementation of the Observer pattern

The events indicate state changes

Your Listener implementations are the “observing” code

1. How do you implement a Listener?

Create a class to implement one or more Listener interfaces

Implement all methods of the interface

Add annotation/register in the web-app tag

1. How does Java EE handle sessions?

When a new session needs to be started, the container creates an HttpSession object, and assigns to it a jsessionid

This ID is carried back to the client in a cookie in the response

The client’s subsequent requests include this ID – the container can use it to retrieve the associated session

JSP

1. What is a JSP?

A JSP is another type of web app component

Written as an HTML-based page with various types of embedded instructions for dynamic functionality

1. What are the lifecycle steps of a JSP?

JSPs have a similar lifecycle to servlets:

* 1. A client requests a JSP page ([**www.example.com/index.jsp**](http://www.example.com/index.jsp))
  2. The JSP engine translates the page into a Java servlet (**index\_jsp.java**)
  3. The Java servlet file is compiled (**index\_jsp.class**)
  4. The web container loads the servlet, instantiates it and calls the **jspInit()** method
  5. **\_jspService()** can now be called to service this request *and* all subsequent requests to this JSP
  6. When **\_jspService()** is invoked, an output HTML file is generated and sent back to the client’s browser

1. List 3 JSP tags
   1. Scriptlets: **<% code %>**
   2. Expression: **<%= code %>**
   3. Declarative: **<%! code %>**
   4. Directive: **<%@ code %>**
2. EL

Expression Language provides a way to retrieve data without the use of expression tags.

${code}

Allows:

Maps,lists, arrays,JavaBean Convention

1. List 3 types of scopes

ApplicationScope

SessionScope

RequestScope

PageScope

1. JSTL

JSP standard Tag Library

<c:if>

<c:out>

1. How do you set values of variables with the request scope?

${requestScope[currentShare]}

Spring

1. What is Spring?

Spring is an open source framework that sits on top of java.

Spring helps structure whole applications in a consistent,productive manner,encouraging SOLID design.

1. What are the key modules?

Data Access

Web

Core Container

1. What are some of the benefits of Spring?

Lightweight

Inversion of Control

AOP(Aspect Oriented Programming)

Container(contains and manages lifecycle of objects)

Boilerplate codee

1. What are the main stages in the lifecycle of a bean?

Spring core Container manages the lifecycle of the beans.

Creation

Injection

Validation

Registration

Destruction

1. What interfaces are involved during the bean lifecycle?
2. What is Dependency Pull?

The objects asks for the dependency.

1. What is Dependency Injection?

Something outside of the object “pushes” its dependencies into it.

1. Why is Dependency Injection better than Dependency Pull?

Low coupling

Testability

1. Difference between IOC DI
   * DI is a form of IOC. Dependencies are passed into an object from the framework, rather than the object having to get the dependencies itself.
2. What are the different ways to declare a bean in Spring?

@Bean in Configuration file

@Component(or @Repository @Controller @Service) in the class

1. What is the purpose of the ApplicationContext?

Used to instantiate the core container and load up the context.xml

1. How do you inject beans into beans?

@Autowired

* + Constructor injection
    - Good for mandatory dependencies
  + Setter injection
    - Good for optional dependencies
  + Field injection
    - This is less desirable due to breaking immutability (doesn't support final fields) and high coupling to the DI framework (if no constructor/setters provided)

1. What is scope?
2. What are some of the ambiguities and issues which can occur?

No beans meet

More than one beans meet

1. How do you solve the ambiguities and issues?

- Create the bean

- @Primary @Qualifier("myQualifier")

Spring MVC

1. What is Spring MVC

Spring web framework

Implements the Front Controller pattern (Java EE pattern)

* + All requests are addressed to a central servlet
  + The servlet delegates requests to controller classes
  + Controllers do the request handling work
  + In Spring, this “central servlet” is known as the DispatcherServlet

Facilitates working with Servlets and JSP

* + No need to create a Servlet for each command

1. Benefits

Encourages use of best practices and SOLID

Separation of concerns

* + Each role (controller, validator, servlet, view, model objects, etc.) can be fulfilled by a specialized object.

Simplification

* + Easy to configure.
  + Far fewer servlet mappings.

Form binding

* + Allows us to bind an object to a Spring form directly.
  + Spring populates the object with form data automatically, instead of having to retrieve form parameters one by one.

1. How does Spring retrieve the correct view to use?

View Resolver

1. What processes user requests?

Controller

1. Where do all requests initially go?

Front Controller DispatcherServlet

1. How do you pass attributes back to the users browser?

Model model

Model.addAttribute()

This will become an attribute in the request scope, accessible by **${attributeName}** or **${requestScope.attributeName}**

Or Spring Form using modelAttribute=

1. Describe the route taken between an initial request and the final response.

