

# DBMS: Triggers & JDBC

Akhilesh Arya

# Triggers

- A trigger is a set of SQL statements that reside in system memory with unique names
- It is a specialized category of stored procedure that is called automatically when a database server event occurs
- Each trigger is always associated with a table

# Cont..

- The following are the main characteristics that distinguish triggers from stored procedures:
  - We cannot manually execute/invoked triggers
  - Triggers have no chance of receiving parameters
  - A transaction cannot be committed or rolled back inside a trigger

# Components of trigger

Part Description	Possible	Values
Trigger Timing	When the trigger fires in relation to the triggering event	BEFORE AFTER
Triggering Event	Which data manipulation operation on the table or view causes the trigger to fire.	INSERT DELETE UPDATE
Trigger Type	How many times the trigger body executes	Statement Row
Trigger Body	What action the trigger performs	Complete PL/SQL block

# Syntax:

```
create trigger [trigger_name]  
[before | after] (timing)  
[insert | update | delete] (event)  
on [table_name]  
[for each row]  
[trigger_body] {SQL statements}
```

- **create trigger [trigger\_name]**: Creates or replaces an existing trigger with the trigger\_name.
- **[before | after]**: This specifies when the trigger will be executed.
- **[insert | update | delete]**: This specifies the DML operation.
- **[table\_name]**: This specifies the name of the table associated with the trigger.
- **[for each row]**: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
- **[trigger\_body]**: This provides the operation to be performed as trigger is fired (SQL statements)

# #Before insert trigger

```
delimiter //  
  create trigger age_verify  
    before insert on customers  
    for each row  
    if new.age<0 then set new.age=0;  
    end if;  
//
```

# #After insert trigger

```
delimiter //  
  create trigger check_null_dob  
    after insert  
    on employees  
    for each row  
    begin  
      if new.birthdate is null then  
        insert into message(message_id, message)  
        values(new.id, concat('Hi', new.name, 'please update your date of  
          birth.'))  
      end if;  
    //
```



# #Before update trigger

delimiter @@

```
create trigger upd_trigger
```

```
before update
```

```
on employees
```

```
for each row
```

```
begin
```

```
if new.salary=10000 then
```

```
    set new.salary=85000;
```

```
elseif new.salary<10000 then
```

```
    set new.salary=72000;
```

```
end if;
```

@@

# #Before delete trigger

```
delimiter //  
create trigger salary_delete  
before delete  
on salary  
for each row  
begin  
insert into salarydel(eid, valid_from, amount)  
values(old.eid,old.valid_from,old.amount);  
end //
```

# ODBC

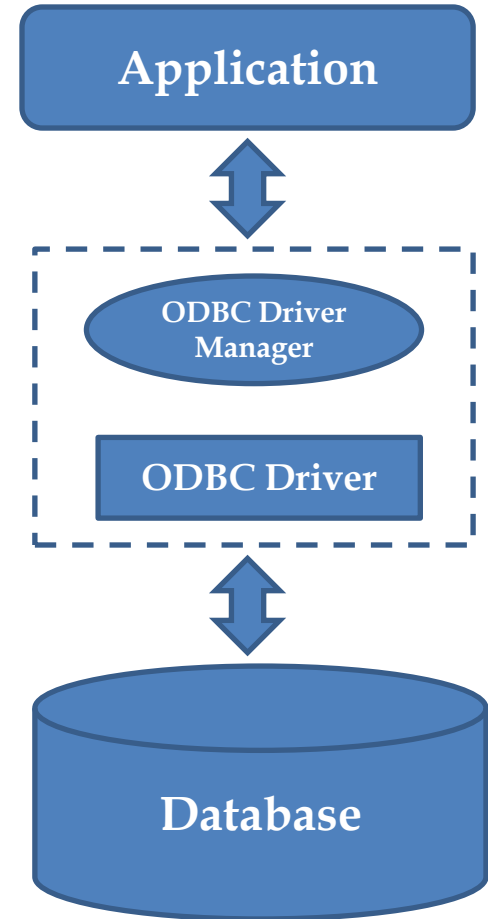
- ODBC stands for **Open Database Connectivity**
- ODBC is an SQL-based Application Programming Interface (API) created by **Microsoft** that is used by Windows software applications to access databases via SQL
- The main reason of using ODBC is its **cross-platform** data access standard power.

# Components of ODBC

There are 4 main components of ODBC these are as follows :

- **Application:**
  - This component basically calls ODBC function and submits SQL statements.
- **Driver Manager:**
  - The role of this component is to load driver for each application.
- **Driver:**
  - Role of this component to handle all function calls and then submits each SQL requests to a data source.
- **Data Source:**
  - Role of this component to access data.

- User submit the request from application to the driver manager
- Driver manager loads driver for each application
- ODBC Driver processes ODBC activity calls, sends SQL requests to a specific data source and returns results in the system.



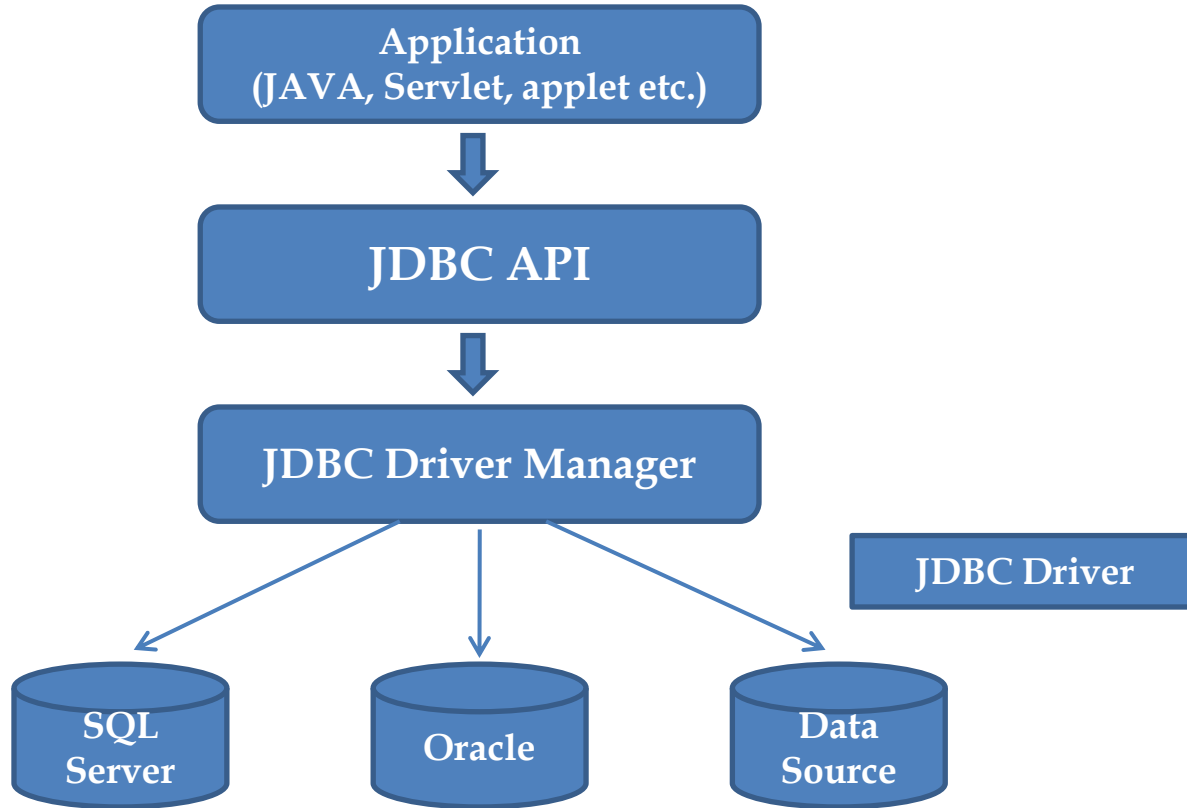
# JDBC

- JDBC is an SQL-based API created by **Sun Microsystems** to enable Java applications to use SQL for database access.
- The JDBC classes are contained in the Java Package *java.sql* and *javax.sql*.

JDBC helps you to write Java applications that manage these three programming activities:

- Connect to a data source, like a database.
- Send queries and update statements to the database
- Retrieve and process the results received from the database in answer to your query

# JDBC Architecture



## Steps to connect JDBC?

1. Import the required package for the corresponding database.
2. Load and register the JDBC drivers.
  - First load then register the same
3. Establish the connection
4. Create a statement
5. Execute the query
6. Process the results
7. Close the connections



ODBC	JDBC
ODBC Stands for Open Database Connectivity.	JDBC Stands for java database connectivity.
Introduced by Microsoft in 1992.	Introduced by SUN Micro Systems in 1997.
We can use ODBC for any language like C,C++,Java etc.	We can use JDBC only for Java languages.
We can choose ODBC only windows platform.	We can Use JDBC in any platform.
Mostly ODBC Driver developed in native languages like C,C++.	JDBC is developed in JAVA language.
For Java applications it is not recommended to use ODBC because performance will be down due to internal conversion and applications will become platform Dependent.	For Java application it is highly recommended to use JDBC because there are no performance & platform dependent problem.
ODBC is procedural.	JDBC is object oriented.