

CHAPTER 8

Module IV

SQL Security

Syllabus

Database Programming with JDBC, Security and authorization:

8.1 Introduction of Views

- Q. What is a view ? How is it created and stored ?
MU - May 12, May 16, 4 Marks
- Q. What is the view in SQL, how it is defined ? Discuss the problem that may arise when we attempt to update a view. How views are implemented ?
MU - May 15, 10 Marks
- Q. Describe view.
MU - May 17, 3 Marks
- Q. Write a short note on : Views in SQL.
MU - May 18, Dec. 18, 5 Marks

1. Definition

- A view is defined as a database object that allows us to create a virtual table in the database whose contents are defined by a query or taken from one or more tables.
- View is defined to hide complexity of query from user.

2. Base table

The table on which view is defined is called as Base table.

3. View - As a window of entire table

- Instead of showing entire table to a user we can show a glimpse of table to the user which is required for him
- **Example :** Consider a student table contains following columns,

STUDENT (Stud_Id, Stud_Name, Std, Div, Addr, Sports, Fees, Cultural_Activity)

- Now for a sports teacher requires only sports related data of students so we can create view called as Stud_Sports_View for teacher as below which will only depicts sports data of student to sports teacher.

- Stud_Sports_View (Stud_Id, Stud_Name, Sports)

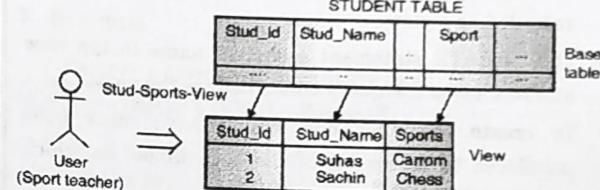


Fig. 8.1.1 : Overview of view

4. Types of views

- Q. What are the types of views ?

- a. Simple view
- b. Complex View

a. Simple view

- The views which are based on only one table called as Simple view.
- Allow to perform DML (Data Manipulation Language) operations with some restrictions.
- Query defining simple view cannot have any join or grouping condition.

b. Complex view

- The views which are based on more than one table called as complex view.
- Do not allow DML operations to be performed.
- Query defining complex view can have join or grouping condition.

5. Working of views

- When we call view in SQL query it refers to database and finds definition of views which is already stored in database.



- Then the DBMS convert this call of view into equal request on the base tables of the view and carries out the operations written in view definition and returns result set to query from which view is called.

8.2 Creating a Views

Q. Explain syntax for creating views.

1. Introduction

- To create a view a subquery must be embedded within the CREATE VIEW statement.
- A simple query is designed and its output can be recorded as a view.
- The CREATE statement assigns a name to the view and also gives the query which defines the view.
- To create the view one should have must have privileges to access all of the base tables on which view is defined.
- Also, the user must have create view permissions from DBA to create a view in the database.
- The create view can change the name of the column in view as per requirements.

2. Syntax

```
CREATE [OR REPLACE] VIEW <view name>
AS
```

- Consider a default HR schema we will create a view of employees having salary below 3000 and retrieve data from view.

SQL> 1 CREATE VIEW EmpBelow3K

```
2 AS
3 SELECT *
4 FROM      Emp
5 WHERE     Sal < 3000;
```

View created.

SQL> SELECT *
 FROM EmpBelow3K;

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	01-MAY-81	2850	30	
7782	CLARK	MANAGER	09-JUN-81	2450	10	
7566	JONES	MANAGER	02-APR-81	2975	20	

SUB QUERY [WITH CHECK OPTION]

- OR REPLACE :** Change the definition of a view without dropping (ALTER VIEW)
- VIEW NAME :** Is name given to a view.
- SUB QUERY :** The query which retrieves the columns of the table that query must have.
- WITH CHECK OPTION :** This is type of check constraint, which specifies that only those rows which are selected by view can be inserted updated or deleted.

3. Example

- In the college database, we may want to let a Head of Departments see only the FACULTY rows for own department.

SQL> CREATE VIEW IT_Faculty

```
AS
SELECT      *
FROM        FACULTY
WHERE       Faculty_Dept = 'IT';
```

/* Selecting Data */

SQL> SELECT *
 FROM IT_Faculty;

7369	SMITH	CLERK	17-DEC-80	800	20	
7499	ALLEN	SALESMAN	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	12-JAN-83	1100		20
7900	JAMES	CLERK	03-DEC-81	950		30
7934	MILLER	CLERK	23-JAN-82	1300		10

11 rows selected.

- Create a view of employee's jobs having salary below 3000 and retrieve data from view.

SQL> CREATE VIEW EmpBelow3K

```

2 AS
3 SELECT DISTINCT(Job)
4 FROM Emp
5 WHERE Sal < 3000;

```

View created.

SQL> SELECT * FROM JobBelow3K;

JOB

```

CLERK
SALESMAN
MANAGER

```

3 rows selected.

8.3 Dropping Views

Q. How to drop view ? Give Syntax.

1. Introduction

- To drop a view we use DROP VIEW statement.
- The DROP VIEW statement requires a name to the view.
- To DROP the view one should have must have privileges to from DBA to DROP a view in database.
- The DROP view dose not affects base table or any column of base table.

2. Syntax

DROP VIEW <View_name> [RESTRICT|CASCADE]

- RESTRICT** : Delete view only if their in no other view depends on this view.

- CASCADE** : Delete view along with all dependent view on original view.

3. Example

Remove a view created in above step.

SQL> DROP VIEW JobBelow3K;

View Dropped;

8.4 Modifying a Views

Q. Give syntax for altering views.

- There are some situations in which some modifications are required in view definition.
- The CREATE OR REPLACE statement in view syntax is used to modify view.
- This statement is used by SQL to overwrite the old view definition with new definition without raising any error like existing view with same name.

Syntax

CREATE OR REPLACE VIEW <View_name>

```

AS
SUB QUERY
[WITH CHECK OPTION]
[WITH READ ONLY];

```

Example

Consider a view defined above 'IT_Faculty' and change it to select all IT faculties having salary above Rs. 25000.

SQL> CREATE OR REPLACE VIEW IT_Faculty

```

AS
SELECT * FROM FACULTY

```



8.4.2 Disadvantages of Views

Q. What are disadvantages of views ?

WHERE	Faculty_Dept = 'IT'
AND	Salary > 25000;

8.4.1 Advantages of Views

Q. What are advantages of views ?

a. Security

- View can restrict user from accessing all data.
- In case of view only data that is given in view is accessible to user. So all data of base table is not accessible to user which will give you security of information.
- For example sports teacher can see data related to sports only and view preventing him from manipulating data pertaining to fees of students.

b. Hides complexity

- The view may be result of very complex query. Hence instead of writing such complicated query again and again we can store such result to a view and access it whenever we want to access.
- So by writing query we can hide the complexity of original query.

c. Dynamic nature

- View definition remains unaffected although there is any change in structure of a table.
- This dynamic nature does not hold true in case if base table is dropped or the column selected by view is altered.
- **Example :** If view is made on two tables, selecting two columns from first table and two columns from second table if we add one more column to first table does not cause any change on view.

d. Does not allows direct access to the tables of data dictionary

- This act like functionality of safeguard to data stored in the data dictionary.
- By this way user cannot change data dictionary to damage database.
- Views can help to make data in data dictionary easily comprehensible and helpful.

e. Data integrity

If data is accessed through a view, the DBMS can automatically check the data to check for specified integrity constraints.

a. Performance

- DBMS translates queries of view to queries on base table.
- Sometimes a simple query may take longer time to run. If view is defined by complex multi table query.
- As the complexity of query is hidden by view hence, users are not aware of how much complicated task the query is actually performing.

b. View management

- The view should be created as per standard then it will be simplifies the job of DBA.
- This happens generally when views are reference other views.
- We need to keep all information of all views in such case so as to it will become very difficult to manage views.

c. Update restrictions

- When a user tries to update a view, the DBMS must translate this query into an update on rows of the underlying base tables.
- Update is possible for simple views.
- Complex views cannot be updated as they are read-only type of views.

8.5 Trigger

Q. What are triggers ? Explain with example.

MU - Dec. 14, May 19, 10 Marks

Q. Describe Trigger.

MU - May 17, 3 Marks

Q. Explain Event Condition Action (ECA) model with suitable example.

MU - Dec. 18, Dec. 19, 10 Marks

1. Introduction

- A trigger is a procedure that is automatically invoked by the DBMS in response to specific alteration to the database or a table in database.
- Triggers are stored in database as a simple database objects.
- A database that has a set of associated triggers is called an active database.

- A database trigger enables DBA (Database Administrators) to create additional relationships between separate databases.

2. Components of Trigger (E-C-A model)

- Event (E)** - SQL statement that causes the trigger to fire (or activate). This event may be insert, update or delete operation database table.
- Condition (C)** - A condition that must be satisfied for execution of trigger.
- Action (A)** - This is code or statement that execute when triggering condition is satisfied and trigger is activated on database table.

3. Trigger syntax

```

CREATE [OR REPLACE] TRIGGER
<Trigger_Name>
[<ENABLE | DISABLE>]
<BEFORE | AFTER>
<INSERT | UPDATE | DELETE>
ON <Table Name>
[FOR EACH ROW]
DECLARE
<Variable_Definitions>;
BEGIN
<Trigger_Code>;
END;

```

OR REPLACE	If trigger is already present then drop and recreate the trigger
<Trigger_Name>	Name of trigger to be created.
BEFORE	Indicates that trigger is to be fired before the triggering event occurs.
AFTER	Indicates that trigger is to be fired After the triggering event occurs.
INSERT	Indicates that trigger is to be fired whenever insert statement adds a row to table.
UPDATE	Indicates that trigger is to be fired whenever Update statement modifies a row in a table.
DELETE	Indicates that trigger is to be fired whenever delete statement removes a row from table.

FOR EACH ROW	Trigger will be fired only once for each row.
WHEN	Contains condition that must be satisfied to execute trigger.
<trigger_code>	Code to be executed whenever triggering event occurs

Fig. 8.5.1 : Trigger parameters

4. Trigger types

a. Row level triggers

- A **row level trigger** is fired each time the table is affected by the triggering statement.
- For example, if an UPDATE statement changes multiple rows in a table, a row trigger is fired once for each row affected by the UPDATE statement.
- If a triggering statement do not affects any row then a row trigger will not run only.
- If FOR EACH ROW clause is written that means trigger is row level trigger

b. Statement level triggers

- A statement level trigger is fired only once on behalf of the triggering statement, irrespective of the number of rows in the table that are affected by the triggering statement
- This trigger execute once even if no rows are affected.
- For example, if a DELETE statement deletes several rows from a table, a statement-level DELETE trigger is fired only one time.
- This is Default type, when FOR EACH ROW clause is not written in trigger that means trigger is statement level trigger

5. Trigger example

- Creating a trigger on employee table whenever new employee added a comment is written to EmpLog Table.
- Let us write a trigger and study its effect.

Example

```

SQL> CREATE OR REPLACE TRIGGER AutoRecruit
2  AFTER INSERT ON EMP
3  FOR EACH ROW

```



```

4 BEGIN
5 Insert into EmpLog values ('Employee
Inserted');
6 END;
7 /

```

Trigger created.

SQL> INSERT INTO EMP

```

2      VALUES  (1,'Mahesh','Manager','1-JAN-
1986',3000,null,10);

```

1 row created.

SQL> SELECT *
 FROM EmpLog;

STATUS

Employee Inserted

Consider other example, whenever there comes a new student add him to CS (Computer Science).

Example

```

SQL> CREATE TRIGGER CSAutoRecruit
      AFTER INSERT ON Student
      FOR EACH ROW
      BEGIN
      INSERT INTO Take VALUES (111, 'CS');
      END;

```

6. Trigger operations

a. Data dictionary for triggers

Once triggers are created their definitions can be viewed by selecting it from system tables as shown below :

Syntax

```

MySQL> SELECT *
      FROM User_Triggers
      WHERE Trigger_Name = '<Trigger_Name>';

```

This statement will give you all properties of trigger including trigger code as well.

b. Dropping Triggers

To remove trigger from database we use command
DROP

Syntax

MySQL> Drop trigger <Trigger_Name>;

c. Disabling Triggers

To deactivate trigger temporarily this can be activated again by enabling it.

Syntax

MySQL> Alter trigger <Trigger_Name> {disable | enable};

7. Trigger advantages

- Triggers are useful for enforcing referential integrity, which preserves the defined relationships between tables when you add, update, or delete the rows in those tables. Make sure that a column is filled with default information.
- After finding that the new information is inconsistent with the database, raise an error that will cause the entire transaction to roll back.

8. Trigger disadvantages

A trigger **hampers the performance of system** as database operations will go on slower due to triggering action.

Restrictions on triggers

- You cannot modify the same table on which triggering event is written.
- You cannot modify a table which is connected to the triggering table by primary key foreign key relation.

Mutating table errors

- This happens when the trigger is querying or modifying table whose modification activated the trigger, or a table that might need to be updated because of a foreign key constraint with a CASCADE policy.
- This problem is called as **mutating table problem**.
- Error** : ORA-04091: table name is mutating, trigger/function may not see it.

9. Solution for mutating table error

The mutating table error is usually the result of a poor application design and mutating triggers should be avoided whenever possible.

- a. **Avoid use of triggers :** The best way to avoid the mutating table error is not to use such triggers. While the Oracle provides methods or procedures that are associated with tables, generally PL/SQL developers avoid triggers unless absolutely necessary.
- b. **Make Use of 'AFTER' trigger :** If use a trigger is must for you, then it is best to avoid the mutating table error by using 'after' trigger, to avoid the currency issues associated with a mutating table. For example, using a trigger "after update on salary", the original update has completed and the table will not be mutating.
- c. **Re-work the trigger syntax :** We can avoid mutating tables with a combination of row-level and statement-level triggers.
- d. **Use autonomous transactions :** You can avoid the mutating table error by marking your trigger as an autonomous transaction, making it independent from the table that calls the procedure.

8.6 Security

- Q. Explain security in SQL. **MU - Dec. 12, 5 Marks**
- Q. Discuss the different security and authorization mechanism in database management system. **MU - May 19, 5 Marks**

1. Introduction

A DBMS system always has a separate system for security which is responsible for protecting database against accidental or intentional loss, destruction or misuse.

2. Threats to Database

a. Confidentiality

- Data in database should be given to only authorized users.
- **Example :** In HR department employee's personnel data should be accessible to that particular employee and the HR person only.

b. Integrity

- Only authorized users should be allowed to modify data.
- **Example :** Only account department can change financial details of company.

c. Availability

- Authorized users can be able to access data any time he wants.
- **Example :** Employee should be able to access own salary any time.

3. Security levels

- a. **Database level :** DBMS system should ensure that the authorization restriction needs to be there on users.
- b. **Operating system level :** Operating system should not allow unauthorized users to enter in system.
- c. **Network level :** Database is at some remote place and it is accessed by users through the network so security is required.

4. Security Mechanisms

a. Access control

- Which identifies valid users who may have any access to the valid data in the Database and which may restrict the operations that the user may perform e.g. ROLE function in SQL.
- **Example :** The movie database might designate two roles : "users" (query the data only) and "designers" (add new data) user must be assigned to a role to have the access privileges given to that role.
- Access privileges are assigned to users and roles.
- Each application is associated with a specified role. Each role has a list of authorized users who may execute / Design / administers the application.

b. Authenticate the user

- Which identify valid users who may have any access to the data in the Database.
- Restrict each user's view of the data in the database.
- This may be done with help of concept of views in relational databases.

c. Cryptographic control / Data encryption

- Encode data in a cryptic form (coded) so that although data is captured by unintentional user still he can't be able to decode the data.
- Used for sensitive data, usually when transmitted over communications links but also may be used to prevent by passing the system to gain access to the data.

**d. Inference control**

- Ensure that confidential information can't be retrieved even by deduction.
- Prevent disclosure of data through statistical summaries of confidential data.

e. Flow control or physical protection

- Prevents the copying of information by unauthorized person.
- Computer systems must be physically secured against any unauthorized entry.

f. Virus control

- At user level authorization should be done to avoid intruder attacks through humans.
- There should be mechanism for providing protection against data virus.

g. User defined control

- Define additional constraints or limitations on the use of database.
- These allow developers or programmers to incorporate their own security procedures in addition to above security mechanism.

8.7 Authorization**Q. Explain Authorization in SQL.****MU - Dec. 12, 5 Marks****Q. Write short notes on : Authorization in SQL.****MU - May 12, May 18, 5 Marks****Q. Discuss the different authorization mechanism in database management system.****MU - May 19, 5 Marks****1. Introduction**

- Authorization is finding out if the person, once identified, is permitted to have the resource.
- Authorization explains that what you can do and is handled through the DBMS unless external security procedures are available.
- This is usually determined by finding out if that person is a part of a particular group, if that person has paid admission, or has a particular level of security clearance.
- Authorization is equivalent to checking the guest list at an exclusive party, or checking for your ticket when you go to the opera.

- Database management system allows DBA to give different access rights to the users as per their requirements.
- In SQL Authorization can be done by using read, insert, update or delete privileges.

2. Basic authorizations

We can use any one form or combination of the following basic forms of authorizations,

a. Resource authorization : Authorization to access any system resource. **Example :** Sharing of database, printer etc.

b. Alteration authorization : Authorization to add attributes or delete attributes from relations.

c. Drop authorization : Authorization to drop a relation.

3. Mandatory access control

- This access control mechanisms can be explained as objects (e.g. tables, views, and rows), subjects (e.g. users, programs), security classes and clearances.
- Each database object is assigned a security class, which define its security parameters and each subject is assigned clearance for a security class;
- Comparing discretionary access control and mandatory access control

4. ROLE Based Access Control (RBAC)

- In **Role-Based Access Control (RBAC)**, access decisions are based on an individual's roles and responsibilities within the organization or user base.
- The process of defining roles is usually based on analysing the fundamental goals and structure of an organization and is usually linked to the security policy.
- For instance, in an educational organization, the different roles of users may include those such as staff, faculty, attendant, student, principal, etc.
- Obviously, these members require different levels of access in order to perform their functions, but also the types of web transactions and their allowed context vary greatly depending on the security policy and any relevant.

5. Discretionary Access Control

- To maintain access rights of each database user discretionary access control is used. It allows multiple users to access some object (i.e. grants the privileges) in a specified mode, like read, write or combination of these.

- Privilege is set of actions that a user can perform on a database object are called the privileges. It is right to execute particular SQL statement on database.
- The high level user (Like DBA) has power to grant access to database and its object.
- This access control is done by two commands GRANT and REVOKE.
- GRANT command helps us to give user privileges to base table and views.

Account creation

- Allows to create a new user account and password for same.
- Allows to create a new user group (Role).

Grant Privilege

DBA can grant some privilege to certain user or role.

Revoke Privilege

DBA can revoke (cancel) all privileges from user or role which are given by him.

8.8 JDBC (Java Database Connectivity)

Q. Explain programming with JDBC.

MU - Dec. 18, 5 Marks

Q. Explain the detailed concept of Database connectivity using JDBC.

1. Introduction

- **Java Database Connectivity (JDBC)** is a java API enables the java programs to execute SQL statements.
- JDBC provides some methods for querying and updating the data in Database Management system such as SQL, Oracle etc.
- JDBC connections which supports creating and executing statements such as SQL INSERT, UPDATE and DELETE.
- Driver Manager is the backbone of the JDBC architecture.

2. Types of JDBC Drivers

A. JDBC Type 1 : JDBC-ODBC Bridge Driver

Q. Write a note on : JDBC-ODBC Bridge driver.

1. Architecture

- Lower level JDBC Driver which makes use of the ODBC driver to connect to the database.
- The driver converts JDBC method calls into ODBC calls and sends them to the ODBC driver.
- This implies that the ODBC driver, as well as the client database code, installed on the client machine.

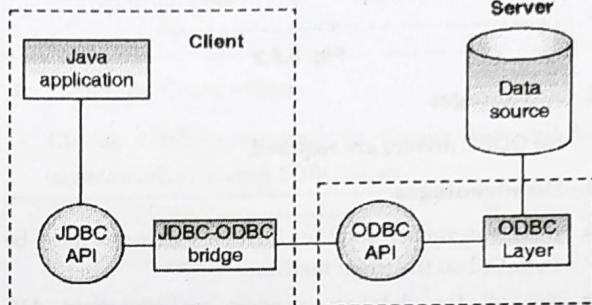


Fig. 8.8.1

2. Advantages

- Many data base contains ODBC drivers so can be implemented generally.
- Easy Installation
- Very useful for which only a small number of clients access to the database.

3. Disadvantages

- Partially written in Java which makes it not portable.
- Not suitable for Web Applications
- Performance overhead as JDBC calls is converted into ODBC calls and then passed on to the ODBC driver.
 - ODBC driver required on the client machine

B. JDBC Type 2 : Native-API/Partly Java Driver

Q. Write a note on : Partly Java driver.

1. Architecture

- Lower-level JDBC Driver API that makes use of the client-side libraries of the database to connect.
- The driver converts JDBC calls into database-specific calls.
- The type 2 driver is not written entirely in Java as it interfaces with non-Java code that makes the final database calls.

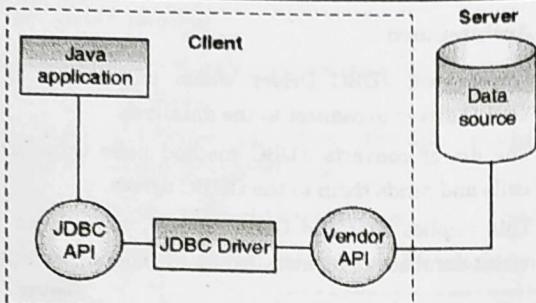


Fig. 8.8.2

2. Advantages

No ODBC drivers are required.

3. Disadvantages

- Database vendor specific client API libraries must be installed on the client machine.
- Not all the database vendors provide client API libraries.
- Not suitable for Web Applications

C. JDBC Type 3 : Network Protocol Driver

Q. Write a note on : Network Protocol driver.

- Known as the Pure Java Driver for database Middleware.
- Three-tiered approach where the client code sends the JDBC calls through the network to a middle-tier server.
- The middle-tier converts the JDBC calls directly or indirectly into database specific protocol, which will forward the JDBC calls to the database.

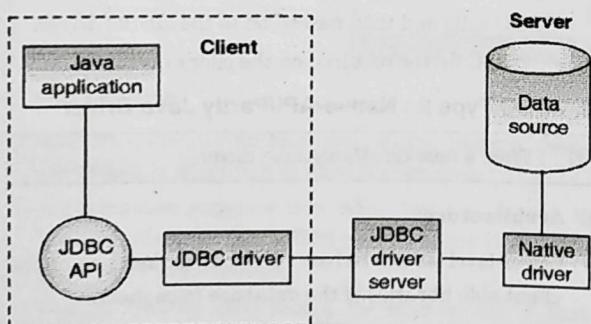


Fig. 8.8.3

1. Advantages

- No database library is required on the client side.
- It can provide additional services like caching, auditing etc.

- Well suited for web applications

2. Disadvantages

The additional layer might be a bottleneck for the JDBC calls.

D) Type 4 : Pure Java Driver

Q. Write a note on : Pure Java driver

- Completely implemented in java
- Driver converts JDBC calls into vendor specific DBMS protocol so that client applications can directly communicate with database server.
- It is also called as thin client driver as all processing at server.
- At client side no processing related to database occurs.

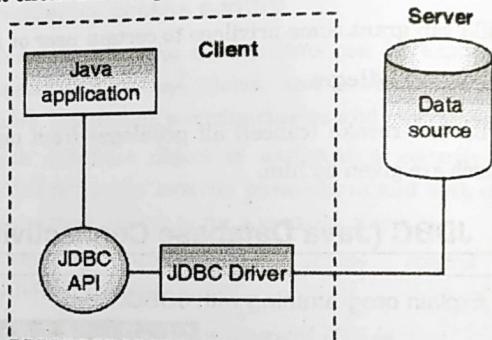


Fig. 8.8.4

1. Advantages

- Platform independent
- Eliminate deployment administration issues.
- Most suitable for web application.
- Very good performance
- No need to install special software on client machine.

2. Disadvantages

- User needs a different driver for each database.
- May not be suitable for some applications if the underlying protocol does not handle issues such as security and network connectivity well.

3. Java Database Connectivity Steps

Q. Explain steps of creating JDBC connection.

1. Define the Connection URL

- Connection URL specifies the database location as well as driver configuration information.

```

String oracleURL = jdbc:oracle:thin:@ hostname :port ;
dbName;
try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver ");
} catch(ClassNotFoundException e)
{
    System.out.println("Error loading driver: " + e);
}

```

2. Establish (open) database the Connection

- Most database applications operate in a client-server environment.
- JDBC applications act as clients and to work with the server they must establish a physical connection to it.
- String username = "User_id";
- String password = "pwd";
- Connection conn
=DriverManager.getConnection(oracleURL,
username, password);

3. Create a Statement Object

- A Statement object is used to send queries and commands to the database.
 - To instantiate a Statement object we call the Connection.createStatement() method.
- ```
Statement stmt = conn.createStatement();
```

## 4. Execute (SQL) Query or Update

- Statement interface provide three method for executing SQL statements. These methods are execute, executeQuery or executeUpdate.
- String query;

```
query = "SELECT Rno, Name, Fee FROM students";
ResultSet rs = stmt.executeQuery(query);
```

## 5. Display the query results

```

while(rs.next())
{
 //Retrieving data by using column name
 int rno= rs.getInt("rno");
 String name = rs.getString("name");
}

```

```

//Retrieving data by using column index
double fee = rs.getDouble(3);
//Display values
System.out.print("RNO: " + rno);
System.out.print(" Name: " + name);
System.out.print(" Fee: " + fee);
}

```

## 6. Close the Connection

- Closing database resources is simple: Just call the close() method of each JDBC object.

```

//Clean-up environment
rs.close();
statement.close();
connection .close();
```

- We have close all ResultSet, Statement, and Connection objects.
- In general, closing the Connection object also closes the ResultSet and Statement object.

## 8.9 University Questions And Answers

### May 2015

- What is the view in SQL, how it is defined ? Discuss the problem that may arise when we attempt to update a view. How views are implemented ? (10 Marks)

### Dec. 2015

- Discuss what is meant by each of the following terms : Trigger

### May 2016

- What is a view ? How is it created and stored ? (4 Marks)

### May 2017

- Describe view and Trigger. (5 Marks)

### May 2018

- Write a short note on : Authorization in SQL (5 Marks)
- Write a short note on : Views in SQL (5 Marks)

**Dec. 2018**

7. Explain programming with JDBC. (5 Marks)
8. Explain Event Condition Action (ECA) model with suitable example (10 Marks)
9. Write a short note on : Views in SQL (5 Marks)

**May 2019**

10. Explain triggers with example. (5 Marks)
11. Discuss the different security and authorization mechanism in database management system. (5 Marks)
12. Explain Event Condition Action (ECA) model with suitable example. (10 Marks)

**Dec. 2019**

□□□