

Group B : Assignment No. 5

Aim : Write a PL/SQL block of code using control structure and Exception Handling

Questions

1. What are the advantages of PL/SQL over SQL.

Ans. PL/SQL is a completely portable, high-performance transaction processing language that offers following advantages :

(1) Support for SQL

PL/SQL lets you use all the SQL data manipulation, cursor control, and transaction control commands, as well as all the SQL functions, operators, and pseudocolumns.

(2) Support for Object-Oriented Programming

By encapsulating operations with data, object types let you move data-maintenance code out of SQL scripts and PL/SQL blocks into methods.

(3) Better performance

With PL/SQL, an entire block of statements can be sent to Oracle at one time. This can drastically reduce communication between your application and Oracle.

④ Full Portability

Applications written in PL/SQL are portable to any operating system

2. List predefined Exceptions.

Ans. PL/SQL declares predefined exceptions globally in package STANDARD.

ACCESS INTO NULL

CASE NOT FOUND

COLLECTION IS NULL

CURSOR ALREADY OPEN

DUP VAL ON INDEX

INVALID CURSOR

INVALID NUMBER

LOGIN DENIED

NO DATA FOUND

NOT LOGGED ON

PROGRAM ERROR

ROWTYPE MISMATCH

SELF IS NULL

STORAGE ERROR

SUBSCRIPT BEYOND COUNT

SUBSCRIPT OUTSIDE LIMIT

SYS INVALID ROWID

TIMEOUT ON RESOURCE

TOO MANY ROWS

VALUE ERROR

ZERO DIVIDE.

GROUP: B ASSIGNMENT NO: 06

AIM: Write a PL/SQL block of code for Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)

PROBLEM STATEMENT:

Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped. Frame the separate problem statement for writing PL/SQL block to implement all types

OBJECTIVES:

- To study cursor programming.
- To learn various cursors attributes.
- To learn different cursor operations.

PRE - REQUISITES:

Interactive SQL commands, PL/SQL programming, use of oracle 11g database Editor.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End :- Oracle Editor
- Back end: Oracle 11g

CURSOR SYNTAX:

A cursor is a temporary work area created in the system memory when a SQL statement is executed. A cursor contains information on a select statement and the rows of data accessed by it.

Group B : Assignment No. 6

Aim : Write a PL/SQL block of code for Cursors .

Questions

1. What are different types of cursor? Explain each type with syntax.

Ans. PL/SQL uses two types of cursors : implicit and explicit.

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed , when there is no explicit cursor for the statement. Whenever a DML statement is issued , an implicit cursor is associated with this statement.

~~Explicit cursors are user-defined cursors for gaining more control over the context area.~~

~~An explicit cursor should be defined in the declaration section of the PL/SQL block. It is created on a SELECT statement.~~

The syntax for creating an explicit cursor is -
CURSOR cursor-name IS select-statement;

2. What are the different attributes of cursor?

Ans. % FOUND

Returns TRUE if an INSERT, UPDATE OR DELETE

statement affected one or more rows or a SELECT INTO statement returned one or more rows.

%NOTFOUND

Returns TRUE if an INSERT, UPDATE or DELETE statement affected no rows, or a SELECT INTO statement returned no rows.

%ISOPEN

Always returns FALSE for implicit cursors.

Returns true if an explicit cursor is OPEN.

%ROWCOUNT

Returns the number of rows affected by an INSERT, UPDATE or DELETE statement, or returned by a SELECT INTO statement.

3. What is parameterized cursor?

Ans. The parameterized cursors are further extension to the explicit cursors having IN type parameters for limiting the number of rows processed by the cursor associated SELECT statement while opening them.

Syntax:

CURSOR cursor-name (parameters) IS
select-statement;

GROUP: B ASSIGNMENT NO: 07

AIM: Write a PL/SQL block of code for Stored Procedure and Stored Function.

PROBLEM STATEMENT:

Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is ≤ 1500 and ≥ 990 then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher Second Class Write a PL/SQL block for using procedure created with above requirement.

Stud_Marks(name, total_marks)

Result(Roll,Name, Class)

OBJECTIVES:

- To learn stored procedure in PL/SQL.
- To learn stored functions in PL/SQL.

PRE - REQUISITES:

Interactive SQL commands, PL/SQL programming, use of oracle 11g database Editor.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End :- Oracle Editor
- Back end: Oracle 11g

Group B : Assignment No. 7

Aim : Write a PL/SQL block of code for stored procedure and stored function.

Questions

1. What is a Stored Procedure?

Ans. A procedure is a subprogram unit that consists of a group of PL/SQL statements. Each procedure has its own unique name by which it can be referred. This subprogram unit is stored as a database object. Call to these procedures can be made by referring to their name, to execute PL/SQL statements.

The values can be passed into the procedure or fetched from procedure through parameters.

Syntax :

~~CREATE OR REPLACE PROCEDURE~~

~~<procedure-name> (<parameters>)~~

~~[IS | AS]~~

~~<declaration part>~~

BEGIN

:

END;

2. Explain the use of %ROWTYPE and %TYPE in SQL.

Ans. The %ROWTYPE attribute is used to provide

a record type that represents a row in a table (or view). Columns in a row and corresponding fields in record have same names and datatypes. However, fields in %ROWTYPE record do not inherit the NOT NULL column constraint.

The %TYPE attribute provides the datatype of a variable or database column. The %TYPE attribute is particularly useful when declaring variables that refer to database columns. %TYPE variables do not inherit the NOT NULL column constraint.

3. Explain IN, OUT and IN-OUT mode in stored procedure.

Ans. IN Parameter

This parameter is used for giving input to the subprograms. It is a read-only variable inside the subprograms. In the calling statement, these parameters can be a variable or a literal value or an expression. By default, the parameters are of IN type.

OUT Parameter

This parameter is used for getting output from the subprograms. It is a read-write variable. In calling statement, these parameters should always be a variable.

IN-OUT Parameter

This parameter is used for both giving input and getting output from the subprograms. It is a read-write variable inside the subprogram. In the calling statement, these parameters should always be a variable to hold the value from the subprogram.

Q. What is a stored Function?

Ans. Function is a standalone PL/SQL subprogram. Functions have a unique name by which it can be referred. These are stored as PL/SQL database objects. Function use RETURN keyword to return the value. A function should always return a value or raise an exception. Function can also return the value through OUT parameters other than using RETURN.

Syntax:

~~CREATE OR REPLACE FUNCTION
 <function-name> (<parameters>)~~

~~RETURN <datatype>~~

~~[IS | AS]~~

~~<declaration>~~

BEGIN

:

.

END;

5. What is the difference between stored Functions and stored procedures?

Ans.	Procedure	Function
i)	Used mainly to execute certain process.	Used mainly to perform some calculation.
ii)	Cannot be called in SELECT statement.	A function that contains no DML statements can be called in SELECT statement.
iii)	OUT parameter is used to return a value.	RETURN is used to return a value.
iv)	Not mandatory to return a value.	Mandatory to return a value.
v)	RETURN will simply exit the control from the subprogram.	RETURN will exit control from subprogram and also return a value.

①

GROUP: B ASSIGNMENT NO: 08

AIM: Write a PL/SQL block of code for Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).

PROBLEM STATEMENT:

Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.

OBJECTIVES:

- To study the concept a trigger.
- To learn various Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).

PRE - REQUISITES:

Interactive SQL commands, PL/SQL programming, use of oracle 11g database Editor.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End :- Oracle Editor
- Back end: Oracle 11g

SYNTAX:

Trigger:

- CREATE [OR REPLACE] TRIGGER trigger_name
- {BEFORE | AFTER | INSTEAD OF }
- {INSERT [OR] | UPDATE [OR] | DELETE}
- [OF col_name]

Group B : Assignment No. 8

Aim : Write a PL/SQL block of code for Database Trigger.

Questions:

1. What is a trigger?

Ans. A trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

2. What are the benefits of triggers?

Ans. Following are the benefits of triggers

- i) Generating some derived column values automatically.
- ii) Enforcing referential integrity
- iii) Event logging and storing information on table access.
- iv) Auditing.
- v) Synchronous replication of tables.
- vi) Imposing security authorizations.
- vii) Preventing invalid transactions.

Q3. What are Row triggers and STATEMENT triggers?

Ans. A row trigger is fired each time the table is affected by the triggering statement. If a triggering statement affects no rows, a row trigger is not run.

eg— If an UPDATE statement updates multiple rows of a table, a row trigger is fired once for each row affected by the UPDATE statement.

A statement trigger is fired once on behalf of the triggering statement, regardless of the number of rows in the table that the triggering statement affects, even if no rows are affected.

eg— If a delete statement deletes several rows from a table, a statement level DELETE trigger is fired only once.

Q. What is INSERT, UPDATE and DELETE triggers?

Ans. DML triggers execute when a user tries to modify data through a DML event. They can be either BEFORE or AFTER triggers.

Triggers on DML statements include following triggers—

BEFORE INSERT, AFTER INSERT, BEFORE UPDATE, AFTER UPDATE, BEFORE DELETE & AFTER DELETE.

GROUP: B ASSIGNMENT NO: 01

AIM: Study of Open Source NOSQL Database: MongoDB (Installation, Basic CRUD operations, Execution).

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Backend: MongoDB

INSTALLATION STEP:

Step 1: First search for mongodb Package:

```
sudo apt-cache search mongodb
```

Step 2: Edit/Create file /etc/apt/sources.list.d/mongo.list

```
sudo vi /etc/apt/sources.list.d/mongo.list
```

OR

```
sudo nano /etc/apt/sources.list.d/mongo.list
```

And past below two lines in mongo.list file

```
##10gen package location  
deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen
```

To save type **ctrl+x** then type Key '**y**' and **Enter**

Step 3: Add GPG Key

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

Step 4: Update package

Group B : Assignment No. 1

Aim : Study of Open Source NoSQL
 Database : MongoDB.

Questions

1. List different NoSQL data models. Explain document based NoSQL data model.

Ans. Different NoSQL data models are

- i) Key-value store
- ii) Document-based store
- iii) Column-based store
- iv) Graph based.

Document-based data model —

The data which is a collection of key value pairs is compressed as a document store.

The values stored (referred to as "documents") provide some structure and encoding of the managed data. XML, JSON, BSON are some common standard encodings. Document store embeds attribute metadata associated with stored content, which essentially provides a way to query the data based on the documents. The fact that document style databases are schema-less makes adding fields to JSON documents a simple task without having to define changes first.

2. Explain Sharding in Mg. MongoDB.

- Ans. i) Sharding is a method for distributing data across multiple machines.
- ii) MongoDB supports horizontal scaling through sharding.
- iii) Horizontal scaling involves dividing the system dataset and load over multiple servers, adding additional servers to increase capacity as required.
- iv) While the overall speed or capacity of a single machine may not be high, each machine handles a subset of the overall workload, potentially providing better efficiency than a single high-speed high-capacity server.
- v) The trade off is increased complexity in infrastructure and maintenance for the deployment.

3. ⚡ What are the features of MongoDB.

- Ans. Some important features of MongoDB are-
- i) High Performance (Indexing)
 - ii) High Availability (Replication)
 - iii) Sharding
 - iv) Aggregation.
 - v) Easy readability
 - vi) Support for multiple storage engines

4. Explain CAP and BASE theorem in NoSQL.

Ans.

The CAP theorem states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:

- Consistency — Every read receives the most recent write or an error.
- Availability — Every request receives a response without guarantee that it contains the most recent write.
- Partition tolerance — The system continues to operate despite an arbitrary number of messages being dropped by network between nodes.

BASE stands for three properties namely:

- Basically Available
- Soft state
- Eventual consistency.

~~Basically Available means that the data in system will mostly be available. The system does not guarantee for data to be always available.~~

Soft state and Eventual consistency goes together. These properties means that the system can be in partially consistent state at any point of time. It will get into consistent state over time if the system doesn't receive any input during that interval.

5. What is Database, Collection and Document in MongoDB?

Ans. Databases —

In MongoDB, databases hold collections of documents.

Collections —

MongoDB stores documents in collections. Collections are analogous to tables in relational databases.

Documents —

MongoDB documents are composed of field and value pairs. MongoDB stores data records as BSON documents.

GROUP: B ASSIGNMENT NO: 02

AIM:

- A. Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators).
- B. Design and Implement any 5 query using MongoDB.
- C. Create simple objects and array objects using JSON.

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Backend: MongoDB

IMPLEMENTATION:

- A. Create Empdb database
- B. Create Employee collection by considering following Fields:
 - i. Empid: Number
 - ii. Name: Embedded Doc (FName, LName)
 - iii. Company Name: String
 - iv. Salary: Number
 - v. Designation: String
 - vi. Age: Number
 - vii. Expertise: Array
 - viii. DOB: String or Date
 - ix. Email id: String
 - x. Contact: String

Assignment No. 2 : Group B.

Questions

1. What is NoSQL and enlist its benefits.

Ans. A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

Benefits of NoSQL include being able to handle:

- i) Large volumes of structured, semi-structured and unstructured data.
- ii) Agile sprints, quick iteration, and frequent code pushes
- iii) Object oriented programming that is easy to use and flexible.
- iv) Efficient, scale-out architecture instead of expensive monolithic architecture.

2. Show relationship of RDBMS terminology with MongoDB.

Ans. The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS	Mongo DB
Database	Database
Table	Collection
Tuple / Row	Document
Column	Field

RDBMS

MongoDB

Table Join

Primary key

Embedded Documents.
-id field.

Database Server and Client

Mysqld / oracle

mongod

mysql / sqlplus

mongo

Q.

3. Explain CRUD operations in MongoDB database with suitable example.

Ans. CRUD operations in MongoDB are as follows -

- Create Operations:

Create or insert operations add new documents to a collection. If the collection doesn't currently exist, insert will create the collection. MongoDB provides the following methods to insert documents into a collection :

db.collection.insertOne()

db.collection.insertMany()

In MongoDB, insert operations target a single collection. All write operations in MongoDB are atomic on the level of single document.

e.g. db.users.insertOne({ name : "sue",
age : 26,
status : "pending" })

- Read Operations

Read operations retrieves documents from a collection ; i.e queries a collection for documents. MongoDB provides the following methods to read from a collection.

`db.collection.find()`

You can specify query filters or criteria that identify the documents to return.

eg- `db.users.find()`

`{ age : { $gt: 18 } }`, ← query criteria

`{ name : 1, address: 1 }` ← projection

`).limit(5)` ← cursor modifier.

- Update Operations

Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection.

`db.collection.updateOne()`

`db.collection.updateMany()`

~~`db.collection.replaceOne()`~~

In MongoDB update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

eg- `db.users.updateMany()`

`{ age : { $lt : 18 } }` ← update filter

`{ $set : { status : "reject" } }` ← update action

)

- Delete operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection.

db.collection.deleteOne()

db.collection.deleteMany()

In mongoDB, delete operations target a single collection.

eg- db.users.deleteMany(

{status: "reject"}) ← delete filter.

4. What are the advantages of mongoDB over RDBMS?

Ans. i) Schema less: MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.

- ii) Structure of a single object is clear.
- iii) No complex joints.
- iv) Deep query ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- v) Tuning.
- vi) Ease of scale out - MongoDB is easy to scale out.
- vii) Conversion /mapping of application objects to database objects not needed.

vii) Uses internal memory for storing the working set, enabling faster access of data.

5. Enlist basic datatypes of MongoDB.

Ans. Following are the basic datatypes of MongoDB.

- i) Double
- ii) String
- iii) Object
- iv) Array
- v) Binary data
- vi) Undefined
- vii) Object Id
- viii) Boolean
- ix) Date
- x) Null
- xi) Regular Expression
- xii) Java Script
- xiii) Symbol
- xiv) Java script with scope.
- xv) Integer.
- xvi) timestamp
- xvii) Min key
- xviii) Max key

6. What is difference between SAVE and UPDATE method?

Ans. Update : modifies an existing document matched

with your query params. If there is no such matching document, that's when upsert comes in picture.

upsert : false — Nothing happens when no such document exists.

upsert : true — New document gets created with contents equal to query params and update params.

Save : Doesn't allow any query-params.
If _id exists and there is a matching document with same _id, it replaces it.
When no _id specified / no matching document, it inserts the document as a new one.

7. What is ObjectID in MongoDB?

Ans. The ObjectID class is the primary key for a mongoDB document and is usually found in _id field in an inserted document.

8. Explain different methods to insert documents.

Ans. To insert a document in MongoDB, you need to use insert() or save() method.

Insert method :

db.collection-name.insert(document)

Save method :

db.collection-name.save(document).

D GROUP: B ASSIGNMENT NO: 03

AIM: Implement aggregation and indexing with suitable example using MongoDB.

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Backend: MongoDB

IMPLEMENTATION:

- A. Use Employee database created in Assignment 01 and perform following aggregation operation
 1. Return Designation with Total Salary is Above 200000
 2. Find Employee with Total Salary for Each City with Designation="DBA"
 3. Find Total Salary of Employee with Designation="DBA" for Each Company
 4. Returns names and _id in upper case and in alphabetical order.
 5. Count all records from collection
 6. For each unique Designation, find avg Salary and output is sorted by AvgSal
 7. Return separates value in the Expertise array where Name of Employee="Swapnil"
 8. Return separates value in the Expertise array and return sum of each element of array
 9. Return Array for Designation whose address is "Pune"
 10. Return Max and Min Salary for each company.

Group B : Assignment No.3

Questions

1. What is MongoDB Aggregation? Explain different types of aggregation methods.

Ans. Aggregation operations process data records and return computed results. Aggregation operations group values from multiple documents together and can perform a variety of operations on the grouped data to return a single result. MongoDB provides three ways to perform aggregation.

i) Aggregation Pipeline

MongoDB's aggregation framework is modeled on the concept of data processing pipelines. Documents enter a multi-stage pipeline that transforms documents into an aggregated result.

ii) Map Reduce

Map reduce operations have two phases: a map stage that processes each document and emits one or more objects for each input document and reduce phase that combines the output of map operation.

iii) Single Purpose Aggregation

MongoDB also provides `db.collection.count()` and `db.collection.distinct()`. All of these operations aggregate documents from a single collection.

2. Enlist different pipeline operators, expression operators and comparison operators.

Ans. Pipeline Operators.

\$project

— Reshapes a document stream.

\$match — Filters the document stream.

\$redact — Restricts the content of a document on a per-field level.

\$limit — Restricts the number of documents

\$skip — Skips a number of documents from pipeline.

\$unwind — Takes an array of documents and returns them as a stream of documents.

\$group — Groups documents together for the purpose of calculating aggregate values.

\$sort — Takes all input documents and returns them in a stream of sorted documents.

\$geoNear — Returns an ordered stream of documents based on proximity to a geospatial point.

* Expression Operators

`$addToSet` — Returns an array of all unique values for the selected field among for each document in that group.

`$first` — Returns the first value in the group.

`$last` — Returns the last value in a group.

`$min` — Returns the lowest value in the group.

`$max` — Returns the highest value in a group.

`$avg` — Returns the average of all values in a group.

`$push` — Returns an array of all values for selected field for each document in that group

~~`$sum` — Returns the sum of all values in a group.~~

* Comparison Operators

`$eq` — Matches values that are equal to specific value

`$gt` — Matches values that are greater than a specified value.

`$gte` — Matches values that are greater than or equal to a specific value.

`$in` — Matches any of the values specified in an array.

`$lt` — Matches values that are less than a specified value.

`$lte` — Matches values that are less than or equal to specified value.

`$ne` — Matches values that are not equal to a specified value.

`$nin` — Matches none of the values specified in an array.

3. Describe SQL to aggregation mapping chart.

Ans. SQL Terms, Functions
and concepts

MongoDB Aggregation
operators

WHERE
GROUP BY
~~HAVING~~
SELECT
ORDER BY
LIMIT
SUM()
COUNT()
join

`$match`
`$group`
`$match`
`$project`
`$sort`
`$limit`
`$sum`
`$sum`
`$sortBy` Count
`$lookup`.

4. Explain indexing methods in Mongo Shell.

Ans. The various indexing methods in Mongo Shell are—

`db.collection.createIndex()` — Builds an index on collection.

`db.collection.dropIndex()` — Removes a specified index on a collection.

`db.collection.dropIndexes()` — Removes all indexes on a collection.

`db.collection.getIndexes()` — Returns an array of documents that describe the existing indexes on a collection.

`db.collection.reIndex()` — Rebuilds all existing indexes on a collection.

~~`db.collection.totalIndexSize()` — Reports the total size used by the indexes on a collection.~~

`cursor.explain()` — Reports on the query execution plan for a cursor.

`cursor_hint()` — Forces MongoDB to use a specific index for a query.

`cursor.max()` — Specifies an exclusive upper index bound for a cursor. For use with `cursor_hint()`.

`cursor.min()` — Specifies an inclusive lower index bound for a cursor. For use with `cursor.hint()`.

5. What are different options for indexing?

Ans. MongoDB provides a number of different index types to support specific types of data and queries.

- i) Single Field Index
- ii) Compound Index
- iii) Multikey Index
- iv) Geospatial Index
- v) Text Index
- vi) Hashed Index.

6. What is the use of Drop Duplicates option in Indexing?

Ans. The use of drop duplicates in indexing is to achieve uniqueness to your index.

7. Write a method to return a list of all indices on a collection and databases.

Ans. To return a list of all indexes on a collection, use the `db.collection.getIndexes()` method.

To list all indexes on all collections in a database, you can use the following operation.

```
db.getCollectionNames().forEach(  
    function (collection) {  
        indexes = db[collection].getIndexes();  
        print("Indexes for " + collection + ":");  
        print json(indexes);  
    });
```

8. Explain different single purpose aggregation operations.

Ans. Count —

MongoDB can return a count of the number of documents that match a query. The count command as well as the count() and cursor.count() methods provide access to count in the mongo shell.

Distinct —

The distinct operation takes a number of documents that match a query and returns all of the unique values for a field in their matching documents. The distinct command and db.collections.distinct() method provide this operation in the mongo shell.

Group —

The group operation takes a number of documents that match a query and then collects groups of documents based on the value of a field or fields. It returns an array of documents, with

computed results for each group of documents.
To Access the group functionality via the
group command or the db.collection.group()
method in the mongo shell.



PART: C ASSIGNMENT NO: 01

AIM: To write a program to implement MongoDB database connectivity with PHP/ python/Java to implement Database navigation operations (add, delete, edit etc.) using ODBC/JDBC.

OBJECTIVES:

- To study Java and MongoDB connectivity using any Java application.
- To perform CRUD operations.

PRE-REQUISITES:

- Basics of NoSQL database-Mongodb
- Java and MongoDB server connectivity.
- MongoDB 2.2.3 or later.
- MongoDB-Java-Driver 2.10.1
- JDK 1.6
- Eclipse 4.2

THEORY:

MongoDB is the leading NoSQL database system which has become very popular for recent years due to its dynamic schema nature and advantages over big data like high performance, horizontal scalability, replication, etc. Unlike traditional relational database systems which provide JDBC-compliant drivers, MongoDB comes with its own non-JDBC driver called Mongo Java Driver. That means we cannot use JDBC API to interact with MongoDB from Java. Instead, we have to use its own Mongo Java Driver API. The official MongoDB Java Driver providing both synchronous and asynchronous interaction with MongoDB.

Group C : Assignment No. 1

Questions.

1. Write steps for connection to MongoDB database

Ans.

Steps for connection to a MongoDB database.

- i) Adding mongo java driver to the CLASSPATH
- ii) Creating a Mongo Client.
- iii) Creating Credentials
- iv) Connecting to MongoDB database.
- v) Accessing the database.

2. Write installation steps for MongoDB on Ubuntu.

Ans. Steps to install MongoDB on Ubuntu—

- i) Import the MongoDB repository.
- ii) Install the MongoDB packages
- iii) Launch MongoDB as a service.
- iv) Configure and connect MongoDB

3. What are different packages needed to connect Java with MongoDB?

Ans

- i) com.mongodb.client.MongoClient
- ii) com.mongodb.client.MongoDatabase
- iii) com.mongodb.client.MongoCollection
- iv) com.mongodb.MongoCredential
- v) org.bson.Document

4. What are unstructured databases? Enlist advantage of NoSQL database.

Ans. Unstructured data is information that either does not have a pre-defined data model or is not organized in a pre-defined manner.

Advantages—

- i) Data Storage.
- ii) Breadth of functionality
- iii) Support for unstructured text
- iv) Database Administration.
- v) Big data application.
- vi) Economy

5. Explain difference between two-tier and three tier architecture.

Ans.

2 - Tier

3 - Tier

- | 2 - Tier | 3 - Tier |
|--|--|
| i) Complex administration | i) Simple as layers are separated. |
| ii) Less secured as client can talk to database directly | ii) Highly secured. |
| iii) Poor Scalability. | iii) Excellent as requests can be load balanced. |
| iv) Poor performance | iv) Good performance. |
| v) High deployment costs as no granularity exists. | v) Less deployment costs due to builder tools. |

```
    Class.forName("com.mysql.jdbc.Driver").newInstance();
} catch (Exception ex) {
    // handle the error
}
```

15

PART: C ASSIGNMENT NO: 02

AIM: To implement MYSQL/Oracle database connectivity with PHP/ python/Java Implement Database navigation operations (add, delete, edit,) using ODBC/JDBC.

OBJECTIVES:

- To connect to MySQL database with the help of Java program.
- Implement database operations on connected database.

PRE-REQUISITES:

- Basics of relational database-MySQL.
- Mysql 5.7 or later.
- mysql-connector-java_8.0.12
- JDK 1.6
- Eclipse 4.2

THEORY:

MySQL provides connectivity for client applications developed in the Java programming language with MySQL Connector/J, a driver that implements the Java Database Connectivity (JDBC) API.

MySQL Connector/J is a JDBC Type 4 driver. Different versions are available that are compatible with the JDBC 3.0 and JDBC 4.x specifications. The Type 4 designation means that the driver is a pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries.

Group C: Assignment No. 2.

1. Enlist applications where the Mysql Java connectivity might be used.

Ans. Applications —

- i) Customer data
- ii) Sales data
- iii) Enterprise data.

2. Write installation steps for MySQL on Ubuntu.

- Ans:
- i) sudo apt-get update
 - ii) sudo apt-get install mysql-server.

3. Which different packages are needed to connect Java with MySQL?

- Ans.
- i) java.sql.Connection
 - ii) java.sql.DriverManager
 - iii) java.sql.Statement
 - iv) java.sql.PreparedStatement
 - v) java.sql.ResultSet
 - vi) java.sql.SQLException.
 - vii) java.sql.Date.

4. What is Relational Database? Give examples.

Ans. The relational database model uses a

collection of tables to represent both data and relationships among those data. Each table has multiple columns and each column has unique name. Tables are also known as relations.

eg—

Microsoft SQL Server, Oracle Database, MySQL and IBM DB2.

(c)

It is a set of one or more columns whose combined values are *unique* among all occurrences in a given table. A key is the relational means of specifying uniqueness. Some different types of keys are:

Primary key – It is an attribute or a set of attributes of a relation which posses the properties of uniqueness and irreducibility (No subset should be unique).

GROUP: C ASSIGNMENT

AIM:

- A. Write a program to implement MongoDB database connectivity with PHP/ python/Java Implement Database navigation operations (add, delete, edit etc.) using ODBC/JDBC.
- B. Implement MYSQL/Oracle database connectivity with PHP/ python/Java Implement Database navigation operations (add, delete, edit,) using ODBC/JDBC.
- C. Design and develop mini project using part A and B with concept of SDLC life cycle.

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.
- To develop the ability to handle databases of varying complexities.
- To use advanced database Programming concepts.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Back End: MongoDB/MySQL

No writeup

THEORY:

Data Models :

A Data Model is a logical structure of Database. It describes the design of database to reflect entities, attributes, relationship among data, constraints etc.

Data Models Types:

A. Record-based Data Models

- The Relational Model