Date :

Assignment

Problem Statement
Design and implement a database schema for a simple university enrollment system.
1. Create tables for students, courses, and enrollments
2. Enforce constraints (PK, FK, NOT NULL, UNIQUE)
3. Create a view V_student_grades that shows student names, course titles, and grades
4. Create an Index on the students table's name column
5. Create a Synonym for the enrollments table
6. Create a sequence to auto generate student_id
7. Populate the tables with sample data using insert statements
8. Write at least 10 DML queries to demonstrate select, update, delete operations.

Theory

Data Definition Language (DDL)
DDL commands are used to define, create, and modify the structure of database objects. These changes are typically present and are auto-committed, meaning they cannot be undone with a Rollback command.

Create - Used to build new database objects.

Date : _____

Alter - Used to modify an existing object's structure.

Drop - Used to permanently delete an object

Truncate - Used to remove all ~~rows~~ rows from a table very quickly.

Database objects & constraints

Table - The fundamental data storage unit, organized into columns (attributes) and rows (tuples)

View - A virtual table that doesn't store data itself. It's a saved select query.
- Hides complex join behind a simple query
- Can be used to restrict access, showing users only certain colums or rows.

Index - A special data structure that the database uses to find rows much faster
- like an index in the back of a textbook. Instead of scanning every page the db looks up the keyword in the index which points directly to the data's location

Date : _____

Sequence - A db object that generates a list of unique numbers, typically used to create primary keys.

Synonym - An alias or alternative name for an object.

Constraints - These are rules that enforce data integrity.

Primary key - A combination of not null and unique. It uniquely identifies every row in the table, ensuring entity integrity.

Foreign key - Enforces referential integrity. It ensures that a value in one column matches a Primary key value in another table, linking the tables.

Not Null - Ensures a column can never have a null value

UNIQUE - Ensures all values in a column are distinct.

Check - Enforces a custom business rule.

Date : _____

Data Manipulation Language (DML)
DML commands are used to manage and manipulate the data within the objects These operations are part of a transaction and are not permanent until a commit command is issued. They can be undone using Rollback.

Select - The primary command for retrieving data
- from - specifies the table
- where - filters the rows
- Group by - Aggregates rows that have the same value into summary rows.
- having - filters the groups.
- orderby - sorts the final result

Insert - Adds new rows in table.

Update - modifies existing data in table.

Delete - Removes specific row from a table.

Date : _____

Conclusion,

This assignment successfully demonstrated the use of DDL commands to build a relational database schema from scratch, including tables, views, indexes, and sequences. The importance of constraints in enforcing data integrity was highlighted. DML commands were then used to populate and query the database, illustrating how data is manipulated and retrieved.

Date :

Assignment

Problem Statement
1. All types of JOINS
2. Sub-queries in the where, from, select clause
3. correlated sub-queries
4. Queries that retrieve data from the view created

Theory

JOINS

A Join is a mechanism for combining rows from two or more table based on a related column. The logical basis for a join is a cartesian product: every row from the first table is combined with every row from the second. The ON or where clause then filters this massive product down to only the row that match.

INNER JOIN

Returns only the rows that have matching values in both tables (intersection)

LEFT JOIN

Returns all rows from the left table, and only the matched rows from the right table.

Date : _____

RIGHT JOIN
The opposite of a left join. Returns all rows from the right table and only the matched rows from the left table.

FULL JOIN
Returns all rows when there is a match in either the left or right table. It's the union of the two sets

CROSS JOIN
Returns the full cartesian Product. It has no ON clause and is rarely used, except for generating test data.

Self Join
A regular join, but where a table is joined to itself. This is used when a table has a internal relationship, such as an employees table where a manager_id column refers back to the employee_id in the same table

Sub Queries (Nested Queries)
A sub-query is a select statement that is nested inside another select, insert, update, delete statements

Date :

In where clause.
- single row sub query (=, >, <)
- Multi row sub query (in, any, all)

In from clause.
Sub query result is treated as a temporary table that the outer query can select from.

In select clause.
The subquery must return a single value, which becomes a new column in the outer query result set.

Conclusion.

This assignment focused on advanced data retrieval techniques. Joins are essential for combining data from a normalized database. Sub queries provide a powerful and flexible way to perform complex, multi-step logical operations and answer sophisticated questions.

Date :

Assignment

Problem statement
Design a mongo DB database bookstore with a collection books. Each document in the books collection should store details like title, author, category, price, stock, and an array of tags Perform CRUD (Create, Read, update, delete) operation.

Theory

Mongo DB is a no SQL, document-oriented database. It contrasts with relational db (RDBMS) in key ways:

RDBMS (SQL)
Rigid schema defined in advance. store data in rows.

Mongo DB (no SQL)
Flexible schema. stores data in JSON like BSON Binary JSON document BSON extends JSON with more data types, like object ID, date, and binary data.

Key concepts

Database
A container for collections.

Date : _____

Collection
A group of related documents. Analogous to a table in SQL

Document
A single record, composed of key-value pairs
Analogous to a Row in SQL

_id field.
Every document must have a unique _id field. If you don't provide one, mongoDB automatically generates a unique objectId. This field is always indexed.

CRUD operations

Create (Insert)
insertOne - Insert one document
insertMany - Insert an array of documents

Read.
find. - The primary query method
query - A document that specifies the filter criteria
projection - A document that specifies which fields to return
findOne - Returns only the first document that matches the query.

Date :

Update
updateOne - updates the first document matching the filter.
updateMany - Updates all documents matching the filter.
update operators - set, inc, push, pull, unset
save() - This legacy method was a wrapper. If the doc contained an id that existed, it performed an update. if it didn't it performed an insert.

Delete
deleteOne - Deletes the first document matching the filter.
deleteMany - Deletes all documents matching the filter.

Conclusion

This assignment provided practical experience with mongoDB document model. we learned how to perform the four fundamental operations of data management (CRUD) using the mongo shell, including the use of query filters and update operators to precisely target and modify data.

Date :

Assignment

Problem statement

Using Oracle db, create two tables

1. Borrower (Roll-no NUMBER(5), Name varchar(20), date_of_issue date, name_of_Book varchar(20) status. char(1))
2. Fine (Roll-no number(5), fine_date date, amt Number(8,2))

write an anonymous PL/SQL block that does the following

1. Accept a Roll-no and Name of Book from user.
2. Retrieves the date of issue for that student and book
3. Calculate the number of days the book has been held.
4. Calculate the fine.
5. If a fine is applicable, insert a new record into the fine table
6. Updates the borrower table status from I to R.
7. Handle exceptions

Theory

PL/SQL is oracle's extension to SQL while SQL is declarative language, PL, SQL is a procedural language. it allows you to add programming logic like variables, loops, and conditions directly into the database engine. This is highly efficient as it avoids network round-trips for complex logic.

Date : _____

An anonymous block is a PL/SQL block of code that is not stored as a db object. but it is executed on the fly. it has a standard structure.

Declare
This section is for declaring all the variables, constants, cursors, and user-defined exceptions that will be used in the block.

Begin
This is the main execution part of the block
It contains all the procedural logic and SQL commands
SQL commands - you can run DML directly
select.. into.. - To retrieve data from a query into variables
Control structures
- If - then - elsif - end - if
- Case - when - then - end case
- loop.. end loop; while.. loop.. end loop

Exception
This section is the error-handler. if any error occurs in the begin block, execution immediately stops and jumps to the exception block to find a matching handler.

Date : _____

Pre-defined exception
System level errors, like no_data_found, too_many_rows, or zero_divide.

User-defined Exceptions
Declared in the declare section they are raised explicitly using the raise command.

when others
A catch-all handler for any exceptions not explicitly named.

Conclusion,

This assignment demonstrated how to use PL/SQL anonymous blocks to execute complex business logic. we successfully used variables, control statement, sql commands, and robust exception handling to manage a library fine calculation.

Date :

Assigoment

Problem statement
Using the bookstore database and books collection from books database.
1. Aggregation
2. Indexing

Theory

Aggregation Pipeline
this is mongoDB's primary tool for data aggregation. It's a frame work that passes documents through a pipeline of stages. Each stage transforms the documents and passes the result to the next stage.

Syntax
db.collection.aggregate([{<stage1>}, {<stage2>}, ...])

key stages:
$match - filters the documents, passing only matching ones to the next stage. this is analogous to the where clause in SQL. It's best to use match early in the pipeline to reduce the number of documents processed.

$group - groups documents by a specified key. it then performs aggregate operations on the grouped data.

Date : ____________

$sort - sorts the output documents. 1 for ascending, -1 for descending.

$project - Reshape the output used to include / exclude fields or create new computed fields. Analogous to the select list in SQL

$limit - Restricts the number of documents passed to the next stage.

$unwind - Deconstructs an array field. If a document has an array with 3 elements, unwind will output 3 seperate document one for each element. This is useful for grouping by array elements.

Indexing

Indexes are special data structure that store a small portion of the collection's data in an easy-to traverse form. This dramatically speeds up read operation.

Types

db.collection.createIndex({field:1})
Creates a single-field index

db.collection.createIndex({field:1, field:2:-1})
Create a compound index, the order of fields

Date :

matters significantly. An index on can also be used for queries on just category, but not for queries on just price.

.explain("executionStats")
A command run on a query that shows how mongoDB executed the query. The "winningPlan" plan section shows
- stage: collscan (bad) or IXscan (good)

Trade-off
Indexes are not free. They consume memory and storage, and they slightly slow down write operations because the index must be updated along with the data.

Conclusion

This assignment explored mongoDB's Powerful aggregation pipeline which is essential for data analysis we learned to group, sort, and project data. Furthermore, we demonstrated the critical importance of indexing, showing how a simple createIndex command can drastically improve query performance by allowing the database to scan an index instead of the entire collection.

Date :

Assignment

Problem statement
Using the bookstore database and books collection, implement a map-Reduce operation to achieve the same goal as part of Assignment
Count the number of books in each category.
1. Write the map function
2. Write the reduce function.
3. Execute the map Reduce command and store the results in a new collection.
4. Compare this approach to the Aggregation pipeline equivalent.

Theory

Map Reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results. It was popularized by google for processing massive dataset. In mongoDB, it consist of three main phases.

Map Phase
- A JS function (map) is executed once for each input document that matches the query.
- The map function uses this to refer to the current document being processed
- The function's job to emit (key, value) pairs
- for this problem, the key is the category and the value is 1
- Example map function:

Date :

```
function() {
    emit(this.category, 1);
}
```

Shuffle and sort phase.
- mongo DB collects all emitted key-value pairs and groups them by the unique key
- It creates a list of values for each key.

Reduce phase.
- A JS function (reduce) is executed for each unique key.
- It receives the key and an array of values
- Its job is to reduce this array to a single output value.

```
function (key, values) {
    return Array.sum(values);
}
```

Execution.

db.collection.mapReduce(mapFunction, reduceFunction, {out: "output_collection_name"})
- The out option is required and specifies where to store the results.

Comparison to Aggregation Pipeline:

Date :

While Map-Reduce is a powerful concept, MongoDB Aggregation Pipeline is now the preferred solution for almost all aggregation tasks.

Performance:- The aggregation pipeline run in native C++ code, making it significantly faster and more memory-efficient than map-reduce, which relies on executing JS.

Usability - The pipeline's declarative stage-based approach is often easier to write, debug, and maintain than custom map and reduce functions.

Use case - Map Reduce is now only recommended for very complex aggregation that cannot be expressed by the pipeline.

Conclusion.

This assignment demonstrated the classic map-reduce model for data aggregation. We successfully implemented map and reduce functions to count books by category. We also learned that while map-reduce is a fundamental concept, MongoDB Aggregation Pipeline is generally the more modern, efficient, and recommended tool for the same tasks.

Date :

Assignment

Problem statement
1. An Explicit Cursor.
2. A Parameterized cursor.
3. A cursor for loop
4. An Implicit Cursor.
Write a PL/SQL block that uses a parameterized cursor to merge N_Roll_call into O_Roll_call. if a Roll_no from N_Roll_call already exists in O_Roll_call, it should be skipped.

Theory

A cursor is a pointer to a context area - a private SQL area - that stores the result set of a select statement. It is a mechanism that allows you to process the result set one row at a time, which is necessary for procedural logic.

Implicit cursor
PL/SQL automatically creates an implicit cursor for all DML and select... into statements. We cannot open, fetch, or close it, but we can check its attributes immediately after the SQL statement.

SQL%found
A boolean that is true if the statement affected one or more rows

SQL%Not found
The logical opposite of SQL%found.

Date :

SQL% RowCount
An integer representing the number of rows affected by the statement This is useful for checking if an insert was successful

SQL% isopen
Always false for implicit cursors, as they close automatically.

Explicit Cursor.
you declare an explicit cursor in the declare section to process a multi row select statement.

1. Declare - Define the cursor with a name and a select query
2. Open - Execute the query and populate the context area.
3. Fetch - Retrieve the next row from the result set into variables. The pointer moves to the next row.
4. Loop & exit - Check the cursor's attributes. The loop must exit when no more rows are found.
5. close - Release the memory and resources held by the cursor.

Date :

Cursor for loop

This is a major simplification and is the preferred method for most cursor operation it implicitly handles the open, fetch, exit. when, and close operation.

- for v_record in c_name loop ... endloop;
- The v_record is implicitly declared as a record of the cursor's %RowType you do not need to declare it in the declare section

Parameterized Cursor.

A cursor that accepts parameters, making it reusable

- Declare cursor c_name(i is select .. where col = p_param_name;
- open c_name (v_value)
- This is more efficient that rebuilding the select statement with a new where clause inside a loop

Conclusion.

This assignment provided a comprehensive look at cursor management in PL/SQL we used an explicit cursor in a for loop to iterate through the new data and a parameterized cursor to efficiently check for the existence of data in the master table the use of SQL%Rowcount demonstrated implicit cursor attributes, providing a complete solution for row by row data processing.

Date :

Assignment

Problem Statement

Write a simple web application using Python, and the flask micro framework that connects to a MySQL database.
The application must support full CRUD operation.

1. Create.
2. Read
3. Update.
4. Delete.

Theory

Database Connectivity is the process of allowing a frontend application to communicate with a backend database.

3 tier Architecture.

This is the standard model for web application.

Presentation tier.
(Client)
↕
Application tier.
(Server)
↕
Data Tier.
(Database)

Date : _____

Key components

Library / Driver:
A specific library is needed to talk to the database for Python and mySQL, this is typically mysql-connector-python or PyMySQL

Connection String
The credentials used to log in to the database

Connection object
An active link to the database, established using the connection string. Best practice is to manage this carefully and close it when done.

Cursor object
A temporary workspace created from the connection. It is used to execute SQL queries and fetch results.

Execution workflow and best practices
1. Import the connector library
2. Establish a connection This is often done in a try... finally block to ensure the connection is always closed, even if an error occurs
3. Create a cursor
4. Execute an SQL query using cursor.execute()
SQL injection: this is the most critical security risk It occurs when a user input is directly pasted into a query string.

Date :

5. commit changes for DML, the changes are not saved until you call connection.commit(). This allows you to group multiple statements into a single transaction that either succeeds or fails as a whole.

6 fetch the result for select queries use cursor.fetchall()

7. close the cursor and the connection to free up database resources

Conclusion.

This assignment successfully demonstrated the complete lifecycle of a data-driven web application we built an application that connects a user-facing frontend to a backend database using an application server This architecture is the foundation of modern web development, and it illustrated the critical importance of connection handling, cursor management, and security (preventing SQL injection)