Asgn 2 Create a Banking database as follows:

Branch (branch_name, branch_city, assets)

Customer (c-id, cname, street, city)

Loan (loan_no, branch_name, amount)

Borrower (c-id, loan_no)

Account (account_no , branch_name, balance)

Depositor(c-id, account_no)

1. Create table for above schemas. (It should include table create, drop, alter and update commands)

2. Create a view for listing all names of customer having account (Saving/current) in “Pune” branch . Alter this view for listing all customers from “Pune” branch having balance greater than 20000 Rs. Rename the view. Perform DML (insert, delete and update) operations on views.

3. Create, alter and drop index on customer and depositor table.

4. Create sequence for required columns.

Asgn 3 Design at least 10 SQL queries for suitable database application using SQL DML statements:

Insert, Select, Update, Delete with operators, functions, and set operator.

Note: All 20 queries are mandatory.

1. Insert values into above created tables

2. Insert values into only selected columns (e.g. in Customer table insert values for only street and city)

3. Select all values from Branch and Account tables.

4. Select customer ID and name from customer table.

5. Select names of all customers who have loan account at Mumbai branch.

6. Assume customer have multiple accounts (saving/ cheking) in specific branch. Accordingly have entries into respective table. Now select list of all distinct customer IDs from Depositor table.

7. Update the city of “XYZ” branch.

8. Update (increase) the loan amount of customer whose c-id is “1211” by 100000 Rs.

9. Delete records of all customers having account balance between 30000 Rs and 80000 Rs.

10. Select all customer names where name starts with ‘R’.

11. List all customers who borrowed loan from (“XYZ”, “ABC”, “PQR”) branches. (IN operator)

12. What is the average account balance of “XYZ” branch.

13. Find total number of customer having account in “ABC” branch

14. Find Maximum and Minimum loan amount of “PQR” branch.

15. Find total account balance of “XYZ” branch.

16. List all details of customer, sorted ascending by name and descending by city.

17. List all customer names from “XYZ” and “ABC” using union.

18. List all customer names (allow duplicate values for names) from “XYZ” and “ABC” using union all

19. List all customer details that are having only loan account in “XYZ”. (Emulate intersect)

20. List all customer details that are having only saving account in “ABC” branch (emulate minus)

Assgn 2 : Create a movie database as follows:

Movie (m-id, title, release-date, rank)

Director (d-id, fname, lname, gender)

Direct (m-id, d-id)

Actor (a-id, f_name, l_name)

Cast (m-id, a-id, role)

1. Create table for above schemas. (It should include table create, drop, alter and update commands)

2. Create a view for listing all movies directed by ‘XYZ’ director. Alter this view for listing all movies of ‘XYZ’ director having rank ‘A’. Rename the view. Perform DML (insert, delete and update) operations on views.

3. Create, alter and drop index on Actor and Cast table.

4. Create sequence for required columns.

Asgn 3. Design at least 10 SQL queries for suitable database application using SQL DML statements:

Insert, Select, Update, Delete with operators, functions, and set operator.

Note: All 20 queries are mandatory.

1. Insert values into above created tables

2. Insert values into only selected columns (e.g. in Movie table insert values for only title and release-date)

3. Select all values from Movie and Actor tables.

4. Select m-ID and title from Movie table.

5. Select details of movie directed by “Yash Chopra”.

6. Assume different actors with same first and last name and accordingly have entries into respective table. Now select list of all distinct f_name and l_name from Actor table.

7. Update the release date of movie “Sholay” and add Pay (money offered for his work) column in Actor table.

8. Update (increase) the pay of actor working in “Ram-Lakhan” by 60000 Rs.

9. Delete records of all actors having pay between 40000 Rs and 70000 Rs.

10. Select all movie names where title starts with ‘K’.

11. List all actors from (“Dangal”, “War”, “PK”) movies.(IN operator)

12. What is the average pay of actors casted in “Airlift”.

13. Find total number of actors working in “Lagaan”.

14. Find Maximum and Minimum pay of actor in “PK”.

15. Find total amount spent on salaries of actors of “War”.

16. List all details of movie, sorted ascending by title and descending by release date.

17. List all actor names from “Lagaan” and “Sholay” using union.

18. List all employee names (allow duplicate values for names) from “PK” and “War” using union all

19. List all actor ids casted in “Lagaan” and “PK”. (Emulate intersect)

20. List all actor ids casted in “Dangal” but not in “War” (emulate minus)

Asgn 2. Create an employee database as follows:

Employee (e-id, name, street,city)

Works (e-id, company_name, salary)

Company (company_name, city)

Manages (e-id, dept_name)

1. Create a table for above schemas. (It should include table create, drop, alter and update commands)

2. Create a view for listing all employees working in ‘ABC’ company. Alter this view for listing all employees of ‘ABC’ company having salary greater than 50000 Rs. Rename the view. Perform DML (insert, delete and update) operations on views.

3. Create, alter and drop index on Employee and Company table.

4. Create sequence for required columns.

Asgn 3. Design at least 10 SQL queries for suitable database application using SQL DML statements:

Insert, Select, Update, Delete with operators, functions, and set operator.

Note: All 20 queries are mandatory.

1. Insert values into above created tables

2. Insert values into only selected columns (e.g. in Employee table insert values for only street and city)

3. Select all values from Employee and Company tables.

4. Select employee ID and name from employee table.

5. Select names of all employees who are managing departments.

6. Assume company is situated in many cities and accordingly have entries into respective table. Now select list of all distinct cities (e.g. “Mumbai”) from Company table.

7. Update the address(street, city) of employee whose E-ID is “12345”.

8. Update (increase) the salary of employees working in “Amazon” by 10000 Rs.

9. Delete records of all employees having salary between 20000 Rs and 50000 Rs.

10. Select all company names where city name starts with ‘P’.

11. List all employees from (“Amazon”, “Flipkart”, “Google”) companies.

12. What is the average salary of employees working in “ABC Corp”.

13. Find total number of employees working in “Amazon”.

14. Find Maximum and Minimum salary of employee in “Google”.

15. Find total amount spent on salaries of employee of “ABC Corp”.

16. List all details of company, sorted ascending by company name and descending by city.

17. List all employee names from “Amazon” and “Google” using union.

18. List all employee names (allow duplicate values for names) from “Amazon” and “Google” using union all

19. List all employee details who are managers (Emulate intersect)

20. List all employee details who are not managers (emulate minus)

Asgn 4 Create a Banking database as follows:

Branch (branch_name, branch_city, assets)

Customer (c-id, cname, street, city)

Loan (loan_no, branch_name, amount)

Borrower (c-id, loan_no)

Account (account_no , branch_name, balance)

Depositor(c-id, account_no)

Subquery

1. Find customer details having loan in “HDFC” and “ICICI”.

2. List customer details who are borrower as well as depositor in a branch.

3. Give count of depositors in each branch.

4. Find all customer details having loan amount greater than average balance at “XYZ” branch.

5. Find customer names from each branch who have highest balance.

Join

6. List depositor details having loan account. (Natural join)

7. Find all possible combination of depositors and borrowers (cross join)

8. List all customer details that are not having loan account. (left outer join)

9. List all customer details who borrowed loan (right outer join)

10. Give details of all customers having loan and deposit account. (Emulate full outer join using union)

Asgn 4 Create a movie database as follows:

Movie (m-id, title, release-date, rank, budget)

Director (d-id, fname, lname, gender)

Direct (m-id, d-id)

Actor (a-id, f_name, l_name, Salary)

Cast (m-id, a-id, role)

Sub-query

1. Find all actors from “Sholay” and “K3G”.

2. Find movie title and director name of top 5 movie arranged in descending order in 2019.

3. Find total number of actors casted in each movie.

4. List role of actor having salary greater than 200000.

5. List movie details directed by female director.

6. Find details of actors having maximum salary in each movie.

Join

7. List all movie details directed by “Karan Johar” (natural join)

8. Find all possible combination of roles can be assigned to each actor (cross join)

9. Show all movie details and director details of the movies released in year 2018-19. (left outer join)

10. Give all actors details not casted in “Dangal” (right outer join)

11. List details of movies not staring by “Amitabh Bachchan” and actor details not casted in “Shakunatala Devi” (emulate full outer join using union)

Asgn 4. Create an employee database as follows:

Employee (e-id, name, street,city)

Works (e-id, company_name, salary)

Company (company_name, city)

Manages (e-id, dept_name)

Subquery

1. Find employee details working in “ABC Corp” and “Amazon”.

2. List of all employees details who are managers.

3. Give count of employees working in each company.

4. Find all employees having salary greater than average salary of employees from “Google”.

5. Find employee names who are managing companies from “Mumbai” city.

Join

6. List of all employees details who are managers. (Natural join)

7. Find all possible combination of managers can be assigned to each department (cross join)

8. List of all employees details who are not managers. (left outer join)

9. Assume company with no employees currently working in it(NULL entry). Fecth employee and company details. (right outer join)

10. Give details of all employees working in company located at “Delhi” and “Mumbai” (emulate full outer join using union)

Assignment No # 5

Create table for following schema and insert suitable data into it:

1. Borrower(Rollin, Name, DateofIssue, NameofBook, Status)

2. Fine(Roll_no,Date,Amt)

Write a PL/SQL block of code to perform below operations:-

1. Accept roll_no & name of book from user. (at present assign values to variables in program only)

2. Check the number of days (from date of issue), if days are between 15 to 30 then fine amounts will be Rs 5per day. If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day. (Use Control structure).

3. After submitting the book, change status (column in Borrower table) from I to R. (I-issue and R-return)

4. If condition of fine is true, then details will be stored into fine table.

5. Write exception handling block as per requirement. (User defines and System defined exception)

Assignment No # 6

Create table for following schema and insert suitable data into it:

1. N_RollCall(RollNo,Name, Branch, Year, Attendance_Percentage, Event)

2. O_RollCall(RollNo,Name, Branch, Year, Attendance_Percentage, Event)

Write a PL/SQL block of code to perform below operations:-

1. Write a PL/SQL block using cursor to count total number of rows updated after increasing attendance of students participated in cultural events by 20% . (IMPLICIT Cursor)

2. Write a PL/SQL block to display student details from computer department. (EXPLICIT Cursor)

3. Write a PL/SQL block to show student detail having attendance greater than 60% (Cursor FOR)

4. Write a PL/SQL block of code using parameterized Cursor, which 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 (PARAMETERIZED Cursor)

Assignment No # 7

Create table for following schema and insert suitable data into it:

1. Stud_Marks(RollNo,name, total_marks)

2. Result(Roll,Name, Class)

Write a PL/SQL block of code to perform below operations:-

1. Write a function comp_Grade() to compute grade of student as follows:

- If marks scored by students in examination is <=1500 and marks >=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

2. Write a stored procedure proc_Grade() which will call above created function comp_Grade() and update entries into Result table with roll number, name and class retuned by function.

Assignment No # 8

Create table for following schema and insert suitable data into it:

1. Library(BookNo, name, author, publication, year)

2. Library_Audit(BookNo, name, author, publication, year)

Write a PL/SQL block of code to perform below operations:-

1. Create a row level trigger which will not allow new entry into Library table if year of publication is before 2005.

2. Update Library table and change publication to “Pearson Education” where author is “Korth” and create a statement level trigger to count total number of rows affected from this update.

3. Create a before trigger to keep track of update operation on Library table and add the old value of updated records in Library_Audit table.

4. Create an after trigger to keep track of delete operation on Library table and add the old value of deleted records in Library_Audit table.

Assignment No # 10

1. Create a student collection in MongoDB. Assume suitable fields.

2. Write a MongoDB query to display all the documents in the collection student.

3. Update marks of student “Ram”.

4. Fetch documents having CGPA 9 or branch as Computer.

5. Fetch documents not having branch as “IT”.

6. Fetch documents having CGPA 7 and native city s “Mumbai”.

7. Fetch documents not having CGPA 9 or branch as Computer.

8. Add new document in the student collection using save() method.

9. Remove document of “Sita”

10. Fetch documents having highest CGPA from computer branch.

Assignment No # 11

1. Create a student collection in MongoDB. Assume suitable fields.

2. Write a MongoDB query to display all the documents in the collection student.

3. Find the strength of students for each branch.

4. List branch wise toppers.

5. Find the details of students securing maximum marks in each subject.

6. Find average marks of each subject.

7. Find the details of student securing minimum marks in “TOC”.

8. Create indexes on student’s roll number and PRN.

9. Display all indexes created above

10. Drop index on roll number.

Assignment No # 12

1. Create an employee collection in MongoDB. Assume suitable fields.

2. Insert single document in a collection

3. Insert multiple documents at a time in a collection

4. Display all documents in a collection

5. Increase salary of an employee working in computer department by 10%.

6. Update address of employee named “Ram”.

7. Delete an entry of “Leena”.

8. Find highest salary in “Mechanical” department.

9. Count number of dependents of each employee.

10. Find total number of employees working in the company.

Assignment No # 13

1. Create Employee object using JSON

2. Create an array of employee objects using JSON