**DBMS CYCLE 1**

1. Create the tables based on the following table instance charts. Choose the appropriate data types and be sure to add integrity constraints.

a. Table name: MEMBER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Column\_** | MEMBER\_ | LAST\_ | FIRST\_NAM | ADDRESS | CITY | PHONE | JOIN |
| **Name** | ID | NAME | E |  |  |  | \_ |
|  |  |  |  |  |  |  | DATE |
| **Key** | PK |  |  |  |  |  |  |
| **Type** |  |  |  |  |  |  |  |
| **Null/** | NN,U | NN |  |  |  |  | NN |
| **Unique** |  |  |  |  |  |  |  |
| **Default** |  |  |  |  |  |  | System |
| **Value** |  |  |  |  |  |  | Date |
|  |  |  |  |  |  |  |  |
| **Data** | NUMBER | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | VARCHAR2 | DATE |
| **Type** |  |  |  |  |  |  |  |
| **Length** | 10 | 25 | 25 | 100 | 30 | 15 |  |
|  |  |  |  |  |  |  |  |

b. Table name: TITLE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column\_** | |  | TITLE\_ID | | TITLE | | DESCRIPTION | |  | RATING | | CATEGORY | RELEASE\_ | |
| **Name** | |  |  | |  | |  | |  |  | |  | DATE | |
|  | |  |  | |  | |  | |  |  | |  |  | |
| **Key** | |  | PK | |  | |  | |  |  | |  |  | |
| **Type** | |  |  | |  | |  | |  |  | |  |  | |
| **Null/** | |  | NN,U | | NN | | NN | |  |  | |  |  | |
| **Unique** | |  |  | |  | |  | |  |  | |  |  | |
| **Check** | |  |  | |  | |  | |  | G, PG, R, | | DRAMA, |  | |
|  | |  |  | |  | |  | |  | NC17, NR | | COMEDY, |  | |
|  | |  |  | |  | |  | |  |  | | ACTION, |  | |
|  | |  |  | |  | |  | |  |  | | CHILD, |  | |
|  | |  |  | |  | |  | |  |  | | SCIFI, |  | |
|  | |  |  | |  | |  | |  |  | | DOCUMEN |  | |
|  | |  |  | |  | |  | |  |  | | TARY |  | |
| **Data Type** | |  | NUMBER | | VARCHAR2 | | VARCHAR2 | |  | VARCHAR2 | | VARCHAR2 | DATE | |
|  | |  |  | |  | |  | |  |  | |  |  | |
| **Length** | |  | 10 | | 60 | | 400 | |  | 4 | | 20 |  | |
|  | | |  | |  | |  | |  |  | |  |  | |
|  | | | | | | | | | | | |  | | |
|  | | c. Table name: TITLE\_COPY | | | | | | | | | |  | | |
|  | |  | | |  | |  | |  | | |  | | |
|  | | **Column** | | | COPY\_ID | |  | | TITLE\_ID | | | STATUS | | |
|  | | **Name** | | |  | |  | |  | | |  | | |
|  | | **Key** | | | PK | |  | | PK,FK | | |  | | |
|  | | **Type** | | |  | |  | |  | | |  | | |
|  | | **Null/** | | | NN,U | |  | | NN,U | | | NN | | |
|  | | **Unique** | | |  | |  | |  | | |  | | |
|  | | **Check** | | |  | |  | |  | | | AVAILABLE, | | |
|  | |  | | |  | |  | |  | | | DESTROYED, | | |
|  | |  | | |  | |  | |  | | | RENTED, | | |
|  | |  | | |  | |  | |  | | | RESERVED | | |
|  | | **FK Ref** | | |  | |  | | TITLE | | |  | | |
|  | | **Table** | | |  | |  | |  | | |  | | |
|  | | **FK Ref** | | |  | |  | | TITLE\_ID | | |  | | |
|  | | **Col** | | |  | |  | |  | | |  | | |
|  | | **Data** | | | NUMBER | |  | | NUMBER | | | VARCHAR2 | | |
|  | | **Type** | | |  | |  | |  | | |  | | |
|  | | **Length** | | | 10 | |  | | 10 | | | 15 | | |
|  | |  | | |  | | | |  | | |  | | |

d. Table name: RENTAL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | BOOK\_ |  | MEMBER\_ | COPY\_ | ACT\_RET\_ | EXP\_RET\_ | TITLE\_ |
| **Name** | DATE |  | ID | ID | DATE | DATE | ID |
| **Key** | PK |  | PK,FK1 | PK,FK2 |  |  | PK,FK2 |
| **Type** |  |  |  |  |  |  |  |
| **Default** | System |  |  |  |  | System Date |  |
| **Value** | Date |  |  |  |  | + 2 days |  |
| **FK Ref** |  |  | MEMBER | TITLE\_ |  |  | TITLE\_ |
| **Table** |  |  |  | COPY |  |  | COPY |
| **FK Ref** |  |  | MEMBER\_I | COPY\_ |  |  | TITLE\_ID |
| **Col** |  |  | D | ID |  |  |  |
| **Data** | DATE |  | NUMBER | NUMBER | DATE | DATE | NUMBER |
| **Type** |  |  |  |  |  |  |  |
| **Length** |  |  | 10 | 10 |  |  | 10 |
|  |  | |  |  |  |  |  |

e. Table name: RESERVATION

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | RES\_ | MEMBER\_ | TITLE\_ |
| **Name** | DATE | ID | ID |
|  |  |  |  |
| **Key** | PK | PK,FK1 | PK,FK2 |
| **Type** |  |  |  |
| **Null/** | NN,U | NN,U | NN |
| **Unique** |  |  |  |
| **FK Ref** |  | MEMBER | TITLE |
| **Table** |  |  |  |
|  |  |  |  |
| **FK Ref** |  | MEMBER\_ID | TITLE\_ID |
| **Column** |  |  |  |
| **Data Type** | DATE | NUMBER | NUMBER |
|  |  |  |  |
| **Length** |  | 10 | 10 |
|  |  |  |  |

1. Create the DEPT table based on the following table instance chart. Place the syntax in a script called lab9\_1.sql, then execute the statement in the script to create the table. Confirm that the table is created.

|  |  |  |
| --- | --- | --- |
| **Column Name** | ID | NAME |
|  |  |  |
| **Key Type** |  |  |
|  |  |  |
| **Nulls/Unique** |  |  |
|  |  |  |
| **FK Table** |  |  |
|  |  |  |
| **FK Column** |  |  |
|  |  |  |
| **Data type** | Number | VARCHAR2 |
|  |  |  |
| **Length** | 7 | 25 |
|  |  |  |

1. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need.
2. Create the EMP table based on the following table instance chart. Place the syntax in a script called lab9\_3.sql, and then execute the statement in the script to create the table. Confirm that the table is created.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | ID | LAST\_NAME | FIRST\_NAME | DEPT\_ID |
|  |  |  |  |  |
| **Key Type** |  |  |  |  |
|  |  |  |  |  |
| **Nulls/Unique** |  |  |  |  |
|  |  |  |  |  |
| **FK Table** |  |  |  |  |
|  |  |  |  |  |
| **FK Column** |  |  |  |  |
|  |  |  |  |  |
| **Data type** | Number | VARCHAR2 | VARCHAR2 | Number |
|  |  |  |  |  |
| **Length** | 7 | 25 | 25 | 7 |
|  |  |  |  |  |

1. Modify the EMP table to allow for longer employee last names. Confirm your modification.
2. Confirm that both the DEPT and EMP tables are stored in the data dictionary. (Hint: USER\_TABLES)
3. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and DEPARTMENT\_ID columns. Name the columns in your new table ID, FIRST\_NAME, LAST\_NAME, SALARY , and DEPT\_ID, respectively.
4. Drop the EMP table.
5. Rename the EMPLOYEES2 table to EMP.
6. Add a comment to the DEPT and EMP table definitions describing the tables. Confirm your additions in the data dictionary.
7. Drop the FIRST\_NAME column from the EMP table. Confirm your modification by checking the description of the table.
8. In the EMP table, mark the DEPT\_ID column in the EMP table as UNUSED. Confirm your modification by checking the description of the table.
9. Drop all the UNUSED columns from the EMP table. Confirm your modification by checking the description of the table.

14. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk

15. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my\_dept\_id\_pk.

16. Add a column DEPT\_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

17. Confirm that the constraints were added by querying the USER\_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab10\_4.sql.

18. Display the object names and types from the USER\_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created.

19. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

The HR department wants you to create SQL statements to insert, update, and delete employee data. As a prototype, you use the MY\_EMPLOYEE table, before giving the statements to the HR department.

Insert data into the MY\_EMPLOYEE table.

20. Run the statement in the dbms\_lab\_01.sql script to build the MY\_EMPLOYEE table to be used for the lab.

21. Describe the structure of the MY\_EMPLOYEE table to identify the column names.

22. Create an INSERT statement to add the first row of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | LAST\_NAME | FIRST\_NAME | USERID | SALARY |
| 1 | Patel | Ralph | rpatel | 895 |
| 2 | Dancs | Betty | bdancs | 860 |
| 3 | Biri | Ben | bbiri | 1100 |
| 4 | Newman | Chad | cnewman | 750 |
| 5 | Ropeburn | Audrey | aropebur | 1550 |

23. Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

24. Confirm your addition to the table

25. Write an INSERT statement in a dynamic reusable script file named loademp.sql to load rows into the MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID. Save this script to a file named dbms\_lab\_02.sql.

26. Populate the table with the next two rows of sample data listed in step 3 by running the INSERT statement in the script that you created.

27. Confirm your additions to the table.

28. Make the data additions permanent.

Update and delete data in the MY\_EMPLOYEE table.

29. Change the last name of employee 3 to Drexler.

30. Change the salary to $1,000 for all employees with a salary less than $900.

31. Verify your changes to the table.

32. Delete Betty Dancs from the MY\_EMPLOYEE table.

33. Confirm your changes to the table.

34. Commit all pending changes.

35. Populate the table with the last row of sample data by modifying the statements in the script that you created in step 6. Run the statements in the script.

36. Confirm your addition to the table.

37. Mark an intermediate point in the processing of the transaction.

38. Empty the entire table.

39. Confirm that the table is empty.

40. Discard the most recent DELETE operation without discarding the earlier INSERT operation.

41. Confirm that the new row is still intact.

42. Make the data addition permanent.

DBMS CYCLE 2

**Exercises on the HR schema: Queries**

1. Display details of jobs where the minimum salary is greater than 10000.
2. Display the first name and join date of the employees who joined between 2002 and 2005.
3. Display first name and join date of the employees who is either IT Programmer or Sales Man.
4. Display employees who joined after 1st January 2008.
5. Display details of employee with ID 150 or 160.
6. Display first name, salary, commission pct, and hire date for employees with salary less than 10000.
7. Display job Title, the difference between minimum and maximum salaries for jobs with max salary in the range 10000 to 20000.
8. Display first name, salary, and round the salary to thousands.
9. Display details of jobs in the descending order of the title.
10. Display employees where the first name or last name starts with S.
11. Display employees who joined in the month of May.
12. Display details of the employees where commission percentage is null and salary in the range 5000 to 10000 and department is 30.
13. Display first name and date of first salary of the employees.
14. Display first name and experience of the employees.
15. Display first name of employees who joined in 2001.
16. Display first name and last name after converting the first letter of each name to upper case and the rest to lower case.
17. Display the first word in job title.
18. Display the length of first name for employees where last name contain character ‘b’ after 3rd position.
19. Display first name in upper case and email address in lower case for employees where the first name and email address are same irrespective of the case.
20. Display employees who joined in the current year.
21. Display the number of days between system date and 1st January 2011.
22. Display how many employees joined in each month of the current year.
23. Display manager ID and number of employees managed by the manager.
24. Display employee ID and the date on which he ended his previous job.
25. Display number of employees joined after 15th of the month.
26. Display the country ID and number of cities we have in the country.
27. Display average salary of employees in each department who have commission percentage.
28. Display job ID, number of employees, sum of salary, and difference between highest salary and lowest salary of the employees of the job.
29. Display job ID for jobs with average salary more than 10000.
30. Display years in which more than 10 employees joined.
31. Display departments in which more than five employees have commission percentage.
32. Display employee ID for employees who did more than one job in the past.
33. Display job ID of jobs that were done by more than 3 employees for more than 100 days.
34. Display department ID, year, and Number of employees joined.
35. Display departments where any manager is managing more than 5 employees.
36. Change salary of employee 115 to 8000 if the existing salary is less than 6000.
37. Insert a new employee into employees with all the required details.
38. Delete department 20.
39. Change job ID of employee 110 to IT\_PROG if the employee belongs to department 10 and the existing job ID does not start with IT.
40. Insert a row into departments table with manager ID 120 and location ID in any location ID for city Tokyo
41. Display department name and number of employees in the department.
42. Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.
43. Display department name and manager first name.
44. Display department name, manager name, and city.
45. Display country name, city, and department name.
46. Display job title, department name, employee last name, starting date for all jobs from 2000 to 2005.
47. Display job title and average salary of employees
48. Display job title, employee name, and the difference between maximum salary for the job and salary of the employee.
49. Display last name, job title of employees who have commission percentage and belongs to department 30.
50. Display details of jobs that were done by any employee who is currently drawing more than 15000 of salary.
51. Display department name, manager name, and salary of the manager for all managers whose experience is more than 5 years.
52. Display employee name if the employee joined before his manager.
53. Display employee name, job title for the jobs employee did in the past where the job was done less than six months.
54. Display employee name and country in which he is working.
55. Display department name, average salary and number of employees with commission within the department.
56. Display the month in which more than 5 employees joined in any department located in Sydney.
57. Display details of departments in which the maximum salary is more than 10000.
58. Display details of departments managed by ‘Smith’.
59. Display jobs into which employees joined in the current year.
60. Display employees who did not do any job in the past.
61. Display job title and average salary for employees who did a job in the past.
62. Display country name, city, and number of departments where department has more than 5 employees.
63. Display details of manager who manages more than 5 employees.
64. Display employee name, job title, start date, and end date of past jobs of all employees with commission percentage null.
65. Display the departments into which no employee joined in last two years.
66. Display the details of departments in which the max salary is greater than 10000 for employees who did a job in the past.
67. Display details of current job for employees who worked as IT Programmers in the past.
68. Display the details of employees drawing the highest salary in the department.
69. Display the city of employee whose employee ID is 105.
70. Display third highest salary of all employees

**Dr. Rajesh K S**