|  |  |  |  |
| --- | --- | --- | --- |
| Cycle No | Course Outcomes | Cycle Problem Topic | Date of Practice |
| 1 | **1**  **1**  **1**  **1**  **1**  **1**  **1**  **1** | Imagine that an organization has many divisions. Each division consists of  many branches and each branch consists of many employees. Draw an ER  Diagram and create the schema for the database objects – **ER Diagram 1** |  |
| 2 | Given the following rules, create an appropriate ERD Diagram and the  schema for the database objects   * A company operates many departments * Each department employs one or more employees * Each of the employee may or may not have one or more dependents * Each employee may or may not have an employee history   **ER Diagram 2** |  |
| 3 | Draw the ER Diagram that models the information requirements in the  following scenario. Create the schema  A manufacturing company has several assembly plants in different cities.  Each plant produces one product that requires certain parts in the assembly.  Parts are from appropriate suppliers located in different cities  – **ER Diagram 3** |  |
| 4 | Use the following business rules to create the ER Diagram. Write all appropriate connections and cardinalities in the ER Diagram   * A department employs many employees but each employee is employed by one department * A division operates many departments * An employee may be assigned many projects. A project may have many employees assigned to it * A project must have at least one employee assigned to it * One of the employee manages each department. Each department is managed by one employee * One of the employees run each division, each division is run by one employee – **ER Diagram 4** |  |
| 5 | College consists of several departments. Department employ teachers and  programme courses. Course is conducted in semesters. Each semester comprises of  several subjects. Students attend the class to learn a subject. Subjects are assessed by  examinations  Given the above rules, create the appropriate Entity Relationship Diagram (ERD) using Chen methodology or Crow’s Foot technique– **ER Diagram 5** |  |
| 6 | Create a database object, TABLE (Organization), with the following attributes and constraints based on ER Diagram 1   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Orgn\_Id | NUMBER | 8 | NOT NULL, PK | Organization Id | | 2 | Orgn\_Name | VARCHAR | 50 | NOT NULL | Organization Name | | 3 | Orgn\_CEO | VARCHAR | 25 | NULL | Organization Chairman | | 4 | Orgn\_Business | VARCHAR | 25 | NOT NULL, CHECK | Type of business of organization | | 5 | Orgn\_Contact | VARCHAR | 25 | NULL | Organization contact person | | 6 | Orgn\_Phone\_No | VARCHAR | 15 | NULL | Organization phone number | |  |
| 6 | Create a database object, TABLE (Division), with the following attributes and constraints   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Divn\_Id | NUMBER | 8 | NOT NULL | Division Id | | 2 | Divn \_Name | VARCHAR | 50 | NOT NULL | Division Name | | 3 | Divn \_Head | VARCHAR | 50 | NOT NULL | Division Head | | 4 | Divn \_Location | VARCHAR | 25 | NOT NULL | City where division is located | | 5 | Divn \_Contact | VARCHAR | 25 | NULL | Division contact person | | 6 | Divn \_Phone\_No | VARCHAR | 15 | NULL | Division phone number | | 7 | Orgn\_Id | NUMBER | 8 | NOT NULL | Organization Id | |  |
| 7 | Create a database object, TABLE (Branch), with the following attributes and constraints. Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Bran\_Id | NUMBER | 8 | NOT NUL | Branch Id. | | 2 | Bran \_Name | VARCHAR | 25 | NOT NULL | Branch Name. City where branch is located | | 3 | Bran \_ Head | VARCHAR | 25 | NULL | Branch Manager’s name | | 4 | Bran \_Total\_Emp | NUMBER | 25 | NOT NULL | Total employees in the branch | | 5 | Bran \_Contact | VARCHAR | 25 | NULL | Branch contact person | | 6 | Bran \_Phone\_No | VARCHAR | 15 | NOT NULL, CHECK | Branch phone number | | 7 | Divn\_Id | NUMBER | 8 | NOT NULL | Division Id | |  | Create PK and FK for the table after defining all columns | | | | | |  |
| 7 | Create a database object, TABLE (Employee), with the following attributes and constraints. Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Emp\_Id | NUMBER | 8 | NOT NULL | Employee Id. PK | | 2 | Emp \_FName | VARCHAR | 25 | NOT NULL | Employee First Name | | 3 | Emp\_LName | VARCHAR | 25 | NOT NULL | Employee Last Name | | 4 | Emp\_Salary | NUMBER | 10,2 | NOT NULL | Employee Salary | | 5 | Emp\_Mgr\_Id | NUMBER | 8 | NULL | Employee Manager Id | | 6 | Emp\_Dept | VARCHAR | 15 | NOT NULL | Employee Department | | 7 | Emp\_Job | VARCHAR | 15 | NOT NULL | Employee Designation | | 8 | Emp\_Join\_Date | DATE |  | NOT NULL | Employee Join Date | | 9 | Emp\_Gender | CHAR | 1 | NOT NULL | Employee Gender | | 10 | Bran\_Id | NUMBER | 8 | NOT NULL | Foreign Key | |  | Create PK and FK for the table after defining all columns | | | | | |  |
| 8 | * Change the structure of the TABLE,Organization to *add* a new column named’Orgn\_Regn\_No’. Data type – VARCHAR, Length 50, NULL * Change the structure of the TABLE,Organization to *add* a new column named ‘Orgn\_Address’. Data type – VARCHAR, Length 50, NULL * Change the structure of the TABLE,Organization to *add* a new column named ‘Orgn\_Web\_Site’. Data type – VARCHAR, Length 50, NULL * Change the structure of the TABLE,Organization to *add* a new column named ‘Orgn\_Email’. Data type – VARCHAR, Length 50, NULL * Change the structure of the TABLE,Organization to *add* a new column named ‘Orgn\_Start\_Date’. Data type – DATE, NOT NULL. Set the default value as sysdate * Change the structure of the TABLE,Organization to *modify* the column named’Orgn\_Business’. Data type – VARCHAR, Length 50, NOT NULL * Change the structure of the TABLE,Organization to *modify* the column named’Orgn\_Reg\_No’. Data type – VARCHAR, Length 30, NOT NULL * Change the structure of the TABLE,Organization to *modify* the column named’Orgn\_Phone\_No’. Data type – NUMBER, Length 15, NOT NULL * Change the structure of the TABLE,Organization to *drop* the column named ‘Orgn\_Web\_Site’ * Change the structure of the TABLE,Organization to *drop* the column named ‘Orgn\_Email’   *Final structure of the table, Organization is as follows*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Orgn\_Id | NUMBER | 8 | NOT NULL, PK | Organization Id | | 2 | Orgn\_Name | VARCHAR | 50 | NOT NULL, UNIQUE | Organization Name | | 3 | Orgn\_Regn\_No | VARCHAR | 30 | NOT NULL | Organization Registration Details | | 4 | Orgn\_CEO | VARCHAR | 25 | NULL | Organization Chairman | | 5 | Orgn\_Business | VARCHAR | 50 | NOT NULL, CHECK | Type of business of organization | | 6 | Orgn\_Address | VARCHAR | 50 | NULL | Organization address | | 7 | Orgn\_Contact | VARCHAR | 25 | NULL | Organization contact person | | 8 | Orgn\_Phone\_No | NUMBER | 15 | NULL | Organization phone number | | 9 | Orgn\_Start\_Date | DATE |  | NULL | Organization start date | |  |
| 9 | * Add a primary key constraint to the table, Division for the column, Divn\_Id * Add a foreign key constraint to the table, Division for the column, Orgn\_Id * Add a UNIQUE constaint to the table, Division for the column, Divn\_Phone\_No * Drop the primary key constraint for the table, Branch * Drop the foreign key constraint for the table, Branch * Add a primary key constraint to the table, Branch for the column, Bran\_Id * Add a foreign key constraint to the table, Branch for the column, Divn\_Id * Add a CHECK constraint to the table, Branch for the column, Bran\_Total\_Emp so that it is greater than zero always * Add a CHECK constraint to the table, Employee for the column, Emp\_Salary so that it is greater than zero always * Add a CHECK constraint to the table, Employee for the column, Emp\_Dept so that user can enter only values ‘Finance', ‘Human Resource’, ‘Purchase’, ‘Logistics’, ‘Sales & Marketing’, ‘Management’, ‘Operation’ * Change the structure of the TABLE,Employee for the column named ‘Emp\_Join\_Date’. Set the default value as sysdate * Add a CHECK constraint to the table, Employee for the column, Emp\_Gender so that user can enter only values ‘M’ or ‘F’ |  |
|  |  |  |
| 9 | Create a database object, TABLE (Organization / Company), with the following attributes and constraints based on ER Diagram 2. *Do not create the table. It has been aleady created*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Orgn\_Id | NUMBER | 8 | NOT NULL, PK | Organization Id | | 2 | Orgn\_Name | VARCHAR | 50 | NOT NULL, UNIQUE | Organization Name | | 3 | Orgn\_Regn\_No | VARCHAR | 30 | NOT NULL | Organization Registration Number | | 4 | Orgn\_CEO | VARCHAR | 25 | NULL | Organization Chairman | | 5 | Orgn\_Business | VARCHAR | 50 | NOT NULL, CHECK | Type of business of organization | | 6 | Orgn\_Address | VARCHAR | 50 | NULL | Organization address | | 7 | Orgn\_Contact | VARCHAR | 25 | NULL | Organization contact person | | 8 | Orgn\_Phone\_No | NUMBER | 15 | NOT NULL | Organization phone number | | 9 | Orgn\_Start\_Date | DATE |  | NOT NULL | Organization start date | |  |
| 10 | Create a database object, TABLE (Department), with the following attributes and constraints based on ER Diagram 2 .   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Dept\_Id | NUMBER | 8 | NOT NULL, PK | Department Id | | 2 | Dept \_Name | VARCHAR | 50 | NOT NULL | Department Name | | 3 | Dept\_Head | VARCHAR | 30 | NULL | Department Head | | 4 | Dept \_ Extn\_No | NUMBER | 8 | NULL | Department Intercomm Number | | 5 | Dept\_ Location | VARCHAR | 50 | NULL | Department Location | |  |
| 11 | * Change the structure of the TABLE,Department to *add* a new column named’Orgn\_Id’. Data type – NUMBER, Length 8, NOT NULL * Add a foreign key constraint to the table, Department for the column, ‘Orgn\_Id’ * Change the structure of the TABLE, Employee to *drop* the column named ‘Emp\_Dept’ * Change the structure of the TABLE,Employee to *add* a new column named’Dept\_Id’. Data type – NUMBER, Length 8, NOT NULL * Add a foreign key constraint to the table, Employee for the column, ‘Dept\_Id’   *Final structure of the table, Department is as follows*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Dept\_Id | NUMBER | 8 | NOT NULL, PK | Department Id | | 2 | Dept \_Name | VARCHAR | 50 | NOT NULL | Department Name | | 3 | Dept\_Head | VARCHAR | 30 | NULL | Department Head | | 4 | Dept \_ Extn\_No | NUMBER | 8 | NULL | Department Intercomm Number | | 5 | Dept\_ Location | VARCHAR | 50 | NULL | Department Location | | 6 | Orgn\_Id | NUMBER | 8 | NOT NULL, FK | Organization Id |   *Final structure of the table, Employee is as follows*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Emp\_Id | NUMBER | 8 | NOT NULL | Employee Id. PK | | 2 | Emp \_FName | VARCHAR | 25 | NOT NULL | Employee First Name | | 3 | Emp\_LName | VARCHAR | 25 | NOT NULL | Employee Last Name | | 4 | Emp\_Salary | NUMBER | 10,2 | NOT NULL | Employee Salary | | 5 | Emp\_Mgr\_Id | NUMBER | 8 | NULL | Employee Manager Id | | 6 | Dept\_Id | NUMBER | 8 | NOT NULL FK | Department Id | | 7 | Emp\_Job | VARCHAR | 15 | NOT NULL | Employee Designation | | 8 | Emp\_Join\_Date | DATE |  | NULL | Employee Join Date | | 9 | Emp\_Gender | CHAR | 1 | NOT NULL | Employee Gender | | 10 | Bran\_Id | NUMBER | 8 | NOT NULL | Foreign Key | |  |
| 12 | Create a database object, TABLE (Dependent), with the following attributes and constraints based on ER Diagram 2. Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Dpen\_Id | NUMBER | 8 | NOT NULL, PK | Dependent Id | | 2 | Dpen \_Name | VARCHAR | 50 | NOT NULL | Dependent Name | | 3 | Dpen\_Relationship | VARCHAR | 30 | NULL | Dependent Relationship | | 4 | Dpen\_Gender | CHAR | 1 | NOT NULL, CHECK | Dependent Gender – ‘M’ or ‘F’ | | 5 | Dpen\_DOB | DATE |  | NULL, Default sysdate | Dependent date of birth | | 6 | Emp\_Id | NUMBER | 8 | NOT NULL, PK,FK | Employee Id | |  | Create a compound PK for the table based on columns, Emp\_Id & Dpen\_Id | | | | | |  |
| 12 | Create a database object, TABLE (Employee History, Emp\_History), with the following attributes and constraints based on ER Diagram 2. Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Emp\_Hist\_Id | NUMBER | 8 | NOT NULL, PK | History Id | | 2 | Emp\_Hist\_Desc | VARCHAR | 50 | NOT NULL | History Description | | 3 | Emp\_Hist\_Date | DATE |  | NOT NULL, Default sysdate | Date of creation of history | | 4 | Emp\_Id | NUMBER | 8 | NOT NULL, PK,FK | Employee Id | |  | Create a compound PK for the table based on columns, Emp\_Id & Emp\_Hist\_Id | | | | | |  |
|  |  |  |
| 13 | Create a database object, TABLE (Doctor), with the following attributes. *(Not related to ER Diagram – Additional exercise)* Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Doct\_Id | NUMBER | 8 | NOT NULL, PK | Doctor Id | | 2 | Doct\_Name | VARCHAR | 50 | NOT NULL | Doctor Name | | 3 | Doct\_Address | VARCHAR | 50 | NULL | Doctor Address | | 4 | Doct\_City | VARCHAR | 50 | NULL | Doctor City | |  | Create PK for the table after defining all columns | | | | | |  |
| 13 | Create a database object, TABLE (Hospital), with the following attributes. *(Not related to ER Diagram – Additional exercise)* Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Hosp\_Id | NUMBER | 8 | NOT NULL, PK | Hospital Id | | 2 | Hosp\_Name | VARCHAR | 50 | NOT NULL | Hospital Name | | 3 | Hosp\_Street | VARCHAR | 50 | NULL | Hospital Street | | 4 | Hosp\_City | VARCHAR | 50 | NULL | Hospital City | |  | Create PK for the table after defining all columns | | | | | |  |
| 14 | Create a database object, TABLE (Visit), with the following attributes. *(Not related to ER Diagram – Additional exercise)* Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Doct\_Id | NUMBER | 8 | NOT NULL, PK | Doctor Id | | 2 | Hosp\_Id | VARCHAR | 50 | NOT NULL,PK | Hospital Id | | 3 | Visit\_Date | DATE |  | NOT NULL, Default Sysdate | Visit Date | |  | Create PK for the table after defining all columns | | | | | |  |
|  |  |  |  |
| 15 | **1** | Create a database object, TABLE (Designation), with the following attributes and constraints based on ER Diagram 5. (***Before proceed with the exercise, ask the lab instructor. Very important***)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Desn\_Id | NUMBER | 8 | NOT NULL, PK | Designation Id | | 2 | Desn \_Name | VARCHAR | 50 | NOT NULL | Designation Name | | 3 | Desn\_Low\_Salary | NUMBER | 8 | NULL | Designation Low Salary | | 4 | Desn\_High\_Salary | NUMBER | 8 | NULL | Designation Low Salary | | 5 | Desn\_Grade | CHAR | 5 | NULL | Designation Grade |   Follow and execute the steps carefully   * Create a database object, TABLE (Designation) with ONLY ONE column named, ‘Desn\_Name’ as follows * CREATE TABLE Designation AS SELECT DISTINCT Emp\_Job AS Desn\_Name FROM Employee * ALTER TABLE Designation ADD (Desn\_Id NUMBER, Desn\_Low\_Salary NUMBER, Desn\_High\_Salary NUMBER, Desn\_Grade CHAR(5) ) * **…………..***See the demo* * Change the structure of the TABLE,Employee to *drop* the column named ‘Emp\_Job’ * Add a foreign key constraint to the table, Employee for the column, Desn\_Id * Add a foreign key constraint to the table, Division for the column, Orgn\_Id * Update the table, Employee for the column, Desn\_Id with the value (Check the Employee data file, ‘Emp\_Data.txt’) * Change the structure of the TABLE,Employee to *add* a new column named’Emp\_Comm’. Data type – NUMBER, Length (8,2) , NULL |  |
|  |  |  |  |
| 15 | **1** | Create a database object, TABLE (Person), with the following attributes. *(Not related to ER Diagram – Additional exercise)* Create the constraints at the *table* level   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | SNo | Column Name | Data Type | Length | Constraint | Remarks | | 1 | Person\_Id | NUMBER | 8 | NOT NULL, PK | Person Id | | 2 | Person \_Name | VARCHAR | 50 | NOT NULL | Person Name | | 3 | Person\_Join\_Date | DATE |  | NOT NULL | Person Join Date | | 4 | Person\_Age | NUMBER | 8 | NOT NULL | Person Age | | 5 | Person\_Marital\_Status | VARCHAR | 1 | NOT NULL | Marital Status | | 6 | Person\_ Gender | VARCHAR | 1 | NOT NULL | Person  Gender | | 7 | Person\_Dept | VARCHAR | 4 | NOT NULL | Department | |  | Create PK for the table after defining all columns | | | | | |  |
| 15 | **1** | * Update the columns, organization contact (person name or receptionist) and start date for all organizations. Year of the organization stat date is to be from 1998 to 2005 * Update division contact (person name) of all divisions whose division ID is from 11 to 15 * Update the branch contact of all branches same as division contact * Update the branch head of all branches whose id is 110 / 131 / 171 * Update the employee join date for all employees. Year of the join date is to be from 2002 to 2015 |  |
| 16 | **1** | 1. Find all organizations which has the business manufacturing  2. Find all organizations which start with the letter 'B' or 'S'  3. List all organizations which started in the year 2010  4. Find out all unique divisions whose organization is doing manufacturing business  5. Find all the divisions whose organization contact is null  6. Find out all the branches belong to a division  7. Find out the total number of employees worked in a particular division  8. List all the branches of all the organizations in ascending order  9. Find out all the employees who belong to a branch |  |
| 16  16 | **1**  **1** | 1. Display Organization Name, Division Name, Branch Name, Employee  Name in descending order  2. Display organization name whose id is 3,4 and 5  3. Display division name whose id is an odd number  4. Display employee names who belong to division 11, 13 and 17  5. Display divisions who does not have any branches at all  6. Display employee names whose name starts with ‘f’ or ‘r’  7. Find out how many employees are there in each branch  8. Find out the employees who belong to a particular company, ‘Microsoft  Corporation’  9. Update the join date for all employees. Year of the date is to be from 1998  to 2010  10. Display all the employees who joined on Jan 01, YYYY  11. Find out the name of employees whose name and branch name starts with  ‘a’  12. Find out how many employees are there in the branch ' Dehradun'  13. Write a query to display the names of employees whose first letter is a  vowel  14. Change the length of the columns EMP\_FNAME and EMP\_LNAME to  60  15. Write a query to find the length of the names of all employees  16. Write a query to generate CSVs from the employee table  17. Get a list of divisions for which no employees are employed  18.Get the list of branches whose divison name is 'Planning'  19.Get the list of divisions where employee with the name 'anju' / ‘ajith’ is  working  20.Generate the list of employees who are working in the company 'Delta  Corporation'  21.Display the list of employees whose first letter of the first name is 'A' and  work in the branch start with the letter 'B'  22. Generate the list of employees along with the division name which is in  ascending order  23. Find out the number of employees in division and its branches  24. Find out the divison name of an employee using CASE...END who works  in the branches 'Baroda' and 'Delhi'  25. Find out the answer for Qn. 24 by using DECODE function |  |
|  | **1** | 1. Find out all doctors who have visited hospitals in same city in which  they live  2. Find which hospital 'Dr. Joshi' has visited  3. Count number of doctors visited to 'Sunrise Hospital and Research  Center' on 26 Novemver 2008  4.Find out how many times 'Dr. Joshi' has visited 'Sunrise Hospital and  Research Center |  |
| 17 | **1** | 1. Get a list of all male employees  2. Get a list of all female employees  3. Get a list of all employees older than 50 years  4. Get a list of all employees younger than 25 years  5. Get a list of all employees between 20 and 30 years of age  6. Get a list of all employees who have joined the company in the year 1995  7. Get a list of all employees from FIN, HRD and COMM departments  8. Get a list of all employees who are not in HRD department  9. Get a list of all employees whose name starts with the letter A  10. Get a list of unmarried male employees greater than 35 years of age,  and unmarried female employees greater than 30 years of age |  |
| 17  17  18  18 | **1**  **1**  **1**  **1** | 1. Create a query to display the last name, designation name, join date and  employee number for each employee who belong to the division from 13  to 20. Employee number appearing first and in ascending order  2. Create a query to display employee name, employee gender, employee  number, employee salary and organization name of the employees who  worked in IT organizations  3. Create a query to find all organization name and register number for which  there are no divisions  4. Create a query to find organization name, division name, branch name and  employee name of those employees who are earning an annual salary of  more than 1,50,000  5. Create a query to display the employee first name, last name, and join date  of those employees who were joined between 01 Jan 2005 and 31 Dec  2010 and who are “Engineers”  6. Create a query to display the divisions under which there are no employees  7. Create a query to display organization name of those employees who have  and ‘a’ or ‘e’ in their last name  8. Create a query to display the employee last name, job role and salary of all  employees who worked in the division 17 and whose salary is more than  12000  9. Create a query to display employee number, first name, last name, salary  and salary increased by 15% and expressed as a whole number. Label the  column as New Salary  10. Create a query that displays the employee’s first name and last name with  the first letter of both capitalized and all other letters lowercase, and the  length of the names, for all employees whose name start with ‘J’, ‘A’, ‘G’  or ‘M’. Give the column an appropriate label  11. Create a query – For each employee, display the employee’s last name  and calculate the number of months between today date and joining date of  the employee. Label the column MONTHS\_WORKED. Order your results  by the the number of months employed. Round the number of months upto  the near whole number  12. Create a query to display the last name and salary for all employees of the  division named 'Web Technologies'. Format the salary to be 15 characters  long, left padded with ‘$’. Label the column SALARY  13. Create a query to display each employee’s last name, join date,  designation name, salary and performance review date which is the first  Monday after six months of service. Label the column REVIEW. Format  the dates to appear in the format similar to “Monday, the Thirty-First of  July, 2016”  14. Create a query to display the last name, join date, day of the week, week  of the month and month on which the employee joined. Label the column  “Date Details”. Order the results by the day of the week starting with  Monday  15. Create a query to find the employee name who are working in branches  with names starting with a vowel  16. Create a query that display department name, employee full name and all  the emplolyees who work in the same department as a given employee.  Give each column an appropriate label  17. Create a query to display the employee name and join date of any  employee who joined after employee ‘AJITH’  18. Create a query to find the highest, lowest, sum and average salary of all  employees. Label the columns as Maximum, Minimum, Sum and  Average respectively. Round your results to the nearest whole number  19. Create a query to show each department’s name, division name, number  of employees, and the average salary for all employees in that department.  Label the column Department, Division, Number of People and Salary  Respectively. Round the average salary to two decimal places    20. Create a query that will display the total number of employees and , of  that total, the number of employees hired in 2005, 2007, 2009 and 2011.  Provide appropriate column headings  21. Create a query to show the employee first name, last name, salary and  department name of those employees in the department 10 and whose  salary is greater than the average salary of the department  22. Create a query to show the employee first name, last name, salary and  department name of those employees whose designation is same as that of  employee with number 1001 and salary is greater than that of employee  with number 1012  23. Create a query to show the department name and minimum salary so that  minimum salary is greater than the minimum salary in department 10  24. Create a query to find the employees who earn the same salary as the  minimum salary for each department  25. Create a quert to show all the data of ‘Managers’ of 'Delta Corporation'  who have been hired after 2007 and before 2011 |  |
| 18 | **1** | 1. Create a query to list the employee first name, last name and department  names of all employees together with the number of years and the number  of completed months that they have been employed and their last name  start with ‘J’, ‘K’, ‘L’ or ‘M’  2. Create a query to show all unique Designations in the department 10, 20  and 30  3. Create a query to show the organization name, division name, branch  name, department name and the number of employees working in each  department that   1. Includes fewer than 3 employees 2. Has the highest number of employees 3. Has the lowest number of employees   4. Create a query to list all employees who were hired on the day of the week  on which the highest number of employees were hired  5. Create a query to show all employee data for ‘Accountant’ and ‘Engineer’  who were joined after the year 2007  6. Create a query to list employee number, first name, last name, department  name, salary and designation name whose last name ends with ‘n’ and  belong to the branch 110 or 171. Create two possible solution  7. Create a query to show the department number and name, and the number  of employees working in each department that   1. Has fewer than 3 employees 2. Has the highest number of employees 3. Has the lowest number of employees |  |
|  |  |  |  |
| 19 | **1**  **1**  **1** | * Create a copy of the following table(s) in your own schema   Organization, Division, Branch, Department, Designation, Employee, Dependent, Doctor, Hospital, Visit and Person. Give name of the table as Orgn\_02, Divn\_02, Branch\_02, Dept\_02, Desn\_02, Emp\_02, Dpen\_02, Doctor\_02, Hosp\_02, Visit\_02 and Person\_02 respectively   * Check the data dictionary tables – USER\_TABLES, USER\_OBJECTS, USER\_CONSTRAINTS and find out the new tables and their constraints * Write the actual name of all constraints of all tables and find the type of constraint of each * Create a sequence named, ‘Orgn\_Seq’ with the following properties – Starting number - 6, Increment -1, Maximum – 10, Minimum - 6, No cycle, No cache * Check the data dictionary table, USER\_SEQUENCES to find out the new sequence created and the current sequence value as well as next sequence value available * Insert data into the table, Orgn\_02 using the sequence, Orgn\_Seq. Provide data only for columns with NOT NULL constraint enabled. (*Create 3 organizations) (Hint – use* * Create a sequence named, ‘Divn\_Seq’ with the following properties – Starting number - 21, Increment -1, Maximum – 25, Minimum - 21, No cycle, No cache * Check the data dictionary table, USER\_SEQUENCES to find out the new sequence created and the current sequence value as well as next sequence value available * Insert data into the table, Divn\_02 using the sequence, Divn\_Seq. Provide data only for columns with NOT NULL constraint enabled. (*Create 5 divisions under the organization with id 3)* * Create an index for the table, Organization based on the columns * Orgn\_Id * Orgn\_Regn\_No   Name the index as ‘Idx\_*Orgn\_Id\_Regn*’   * Check the data dictionary table, USER\_INDEXES to find out the new index created for the table, Organization * Create an index for the table, Division based on the columns * Divn\_Id * Divn\_Name   Name the index as ‘Idx\_*Divn\_Id\_Name*’   * Check the data dictionary table, USER\_INDEXES to find out the new index created for the table, Division * Create an index for the table, Branch based on the columns * Bran\_Id * Bran\_Name   Name the index as ‘Idx\_*Bran\_Id\_Name*’   * Check the data dictionary table, USER\_INDEXES to find out the new index created for the table, Branch * Create an index for the table, Department based on the columns * Dept\_Id * Dept\_Name   Name the index as ‘Idx\_*Dept\_Id\_Name’*   * Check the data dictionary table, USER\_INDEXES to find out the new index created for the table, Department * Create an index for the table, Employee based on the columns * Emp\_Id * Emp\_FName * Emp\_LName   Name the index as ‘Idx\_*Emp\_Id\_Name’*   * Check the data dictionary table, USER\_INDEXES to find out the new index created for the table, Employee * Find out all primary indexes created in the schema by querying the table, USER\_INDEXES * Create a view for the table, Organization based on the columns - Orgn\_Id,   Orgn\_Name, Orgn\_Regn\_No, Orgn\_CEO, Orgn\_Business,  Orgn\_Phone\_No and Orgn\_Start\_Date  Name the view as ‘Vw\_Orgn\_1*’*   * Create a view for the table, Organization which started during the period from 2002 to 2005 based on the columns - Orgn\_Id, Orgn\_Name, Orgn\_Regn\_No, Orgn\_CEO, Orgn\_Phone\_No and Orgn\_Start\_Date   Name the view as ‘Vw\_Orgn\_2*’* |  |
| 20  21 |  | * Wite an anonymous program to insert data into the Organization table using a sequence with Organization ID ranges from 6 to 10 and other details will be provided during run time   (Hint :- Use '&' to provide non Organization ID details)   * Consider the following Organizational Hierarchy   - Chairman and Chief Executive Officer  - Executive Vice President  - Vice President  - Director  - Project Manager  - Project Leader  - Software Engineer  Write an anonymous program to display the above hierarchy in that order using nested anonymous blocks such that one block will display one and only one designation   * Write an anonymous program to insert an already existing data of Division Id into the Division table and catch the error in the exception. Display the message as   "Division Id already exist" to the user     * Write an anonymous program using cursors to display the following details from Company, Department and Employee tables   (Hint :- Use cursors with parameter syntax)  - Company Name  - Department Name  - Employee Name &  - Dependent Name   * Write an anonymous program using FOR..LOOP to display the following marks of a subject in the examination   - Subject Name  - Total marks given  - Average marks  - Minimum marks  - Maximum marks   * Write an anonymous program to store the marks of a student with ID 6 and subject codes from 501 to 508 and marks 72, 65, 58, 80, 60, 83, 59 and 62 respectively. After saving the data, retrieve and display the details of marks   HINT as follows  - Find out whether a student with ID 6 already exist. If exist, inform  the user and exit from the program.  - If not exist, store the data using a FOR..LOOP  - Display the data using a CURSOR  - Update the marks for the subject 502 and 503 with 70 and 60  respectively  - Display the data using a CURSOR   * Write an anonymous program to display the following order details of those customers whose total order amount is greater than 15000   (HINT :- Use cursor)  - Customer Name  - Order ID  - Total Amount   * Write an anonymous program to display the following details using a cursor     - Employee ID  - Employee Name,  - Designation Name  - Designation Fee     * Write an anonymous program to display all of the project's total cost greater than 50000   Arrange the data in the descending order of Project Name    (HINT :- Different staff with different designation and fee is working in a project)    - Project ID  - Project Name  - Total Hours  - Total Cost |  |
| 22-25  22-25  22-25 | **2** | * Write a stored procedure to save the deails of a new organization into the table * Write a stored procedure to display the details of a few organizations (*Hint* :- Pass the organization ID values from range – to range) * Write a stored procedure to save the deails of a new organization using a sequence number * Write a nested stored procedure to create a new organization, update the name of the organization, count the number of rows in the organization table and delete any one organization * Write a stored procedure to display the details of a few divisions (*Hint* :- Pass the division ID values from range – to range) * Write a stored procedure to calculate the tax based on the annual salary of an employee based on his/her number   Tax rate is given as follows  100000 – 150000 -> 5%  150001 – 180000 -> 8%  180001 – 240000 -> 10%  Above 240000 -> 12%   * Write a stored procedure to display the employee number, name and joining date of all employees in the format “Monday, 05TH of March, 2010” * Write a stored procedure to find the monthly salary, quarterly salary, semi annual salary and annual salary of an employee based on the number * Write a stored procedure to find the number of employees in a particular branch based on branch name * Write a stored procedure to insert employee details into a table for a non existing department (*Hint* :- Take instructions from the teacher before proceeding with the program) * Write a stored procedure to get the details of employee number, employee name, dept. number and dept. name without using a join (Hint :- Use IF..THEN ..ELSE..END IF statement) * Write a stored procedure to increase the salary of an employee by 10% and change the designation. The old salary and designation is to be stored in the employee history table * Write a stored function to save the deails of a new organization into the table * Write a stored function to display the details of a few organizations (*Hint* :- Pass the organization ID values from range – to range) * Write a stored function to save the deails of a new organization using a sequence number * Write a nested stored function to create a new organization, update the name of the organization, count the number of rows in the organization table and delete any one organization * Write a stored function to display the details of a few divisions (*Hint* :- Pass the division ID values from range – to range) * Write a stored function to calculate the tax based on the annual salary of an employee based on his/her number   Tax rate is given as follows  100000 – 150000 -> 5%  150001 – 180000 -> 8%  180001 – 240000 -> 10%  Above 240000 -> 12%   * Write a stored function to display the employee number, name and joining date of all employees in the format “Monday, 05TH of March, 2010” * Write a stored function to find the monthly salary, quarterly salary, semi annual salary and annual salary of an employee based on the number * Write a stored function to find the number of employees in a particular branch based on branch name * Write a stored function to insert employee details into a table for a non existing department (*Hint* :- Take instructions from the teacher before proceeding with the program) * Write a stored function to get the details of employee number, employee name, dept. number and dept. name without using a join (Hint :- Use IF..THEN ..ELSE..END IF statement) * Write a stored function to increase the salary of an employee by 10% and change the designation. The old salary and designation is to be stored in the employee history table |  |