Q.SET 1

1. Create an employee table ‘EMP’ with following fields :

empno                NUMBER(2)

ename                 VARCHAR2(25)

job                   VARCHAR2(12)

salary                 NUMBER(10,2)

commission              NUMBER(7,2)

deptno                 NUMBER(2)

1. Display the structure of ‘EMP’
2. Insert the following record into ‘EMP’

**EMPNO ENAME JOB SAL COMM DEPTNO**

7369 SMITH CLERK 800 20

1. Insert the rest of records using substitution variable.

**EMPNO ENAME JOB SAL COMM DEPTNO**

7499 ALLEN SALESMAN 1600 300 30

7521 WARD SALESMAN 1250 500 30

7566 JONES MANAGER 2975 20

7654 MARTIN SALESMAN 1250 1400 30

7698 BLAKE MANAGER 2850 30

7782 CLARK MANAGER 2450 10

7788 SCOTT ANALYST 3000 20

7839 KING PRESIDENT 5000 10

7844 TURNER SALESMAN 1500 30

7876 ADAMS CLERK 1100 20

7900 JAMES NULL 950 30

1. FORD ANALYST 3000 20

7934 MILLER CLERK 1300 10

1. Insert job as ‘CLERK’ for all ‘NULL’ job types.
2. Add a new field ‘date\_join’ with following values

**date\_join**

17-DEC-80

20-FEB-81

22-FEB-81

02-APR-81

28-SEP-81

01-MAY-81

09-JUN-81

19-APR-87

17-NOV-81

08-SEP-81

23-MAY-87

03-DEC-81

03-DEC-81

23-JAN-82

1. Display details of all employees.
2. Display all the distinct job types in ‘EMP’.
3. Display names of all employees in dept 20 and 30
4. List name and Total of salary i.e sal+commission
5. List name and Annual Salary i.e sal\*12
6. List the employee who joined in the date ‘03-DEC-81’
7. Display the total salary of ‘Miller’
8. Delete the employee ‘Miller’ from’EMP’
9. Display name and deptno of all employees.
10. Remove the field ‘commission’ fom’EMP’ after updating salary with total salary, i.e sal+commission
11. Display the name of employees having the same amount of salary **( don’t use subqueries**)
12. Display the name and employee no as ‘name’ and ‘emp\_id’
13. Rename table ‘EMP’ to ‘EMPLOYEE’
14. Create a new table ‘EMP\_TAB’ from table ‘EMPLOYEE’
15. List the details of ‘EMPLOYEE’ and ‘EMPTAB’
16. Delete all records from ‘EMP’
17. Delete the table ‘EMP’

**Q.SET 2**

Create a table STUDENT with fields sid, name, dob (date of birth) and marks of 3 subjects ( physics, chemistry and maths ). Add the details of 5 students. Perform the following queries:

1. Display the id and name of youngest student.
2. Display the details of students who have passed in maths and either in physics or chemistry.(pass mark = 40 marks and above)
3. Add two more columns total and average.
4. Display the name of student who scored highest marks in maths.
5. Display the name of student who scored least marks in chemistry.
6. Update column total with total marks.
7. Display details of students in order of total merit.
8. Rename the column average with avg\_mark
9. Find out the overall average of class.
10. Display details of students whose average is greater than overall average.
11. Find the total no: of students whose average is greater than overall average.

**Q.SET 3**

Create the Table LOAN\_ACCOUNTS with the structure given below

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Length |
| Accno | CHAR | 4 |
| Cust\_name | VARCHAR2 | 15 |
| Loan\_Amount | NUMBER | 7 digits and 2 decimal places |
| Installments | NUMBER |  |
| int\_rate | NUMBER | 2 digits and 2 decimal places |
| Start\_date | DATE |  |
| Interest | NUMBER | 7 digits and 2 decimal places |

Add another column ‘category’ of type varchar2(1) in the Loan Table.

Insert the following details into the table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accno | Cust\_name | Loan\_Amount | Installments | int\_rate | Start\_date | Interest |
| 1001 | R.K Gupta | 300,000.00 | 36 | 12.00 | July 19, 2009 |  |
| 1002 | S.P Sharma | 500,000.00 | 48 | 10.00 | March 22, 2008 |  |
| 1003 | K.P Jain | 300,000.00 | 36 | NULL | August 3, 2007 |  |
| 1004 | M.P Yadav | 800,000.00 | 60 | 10.00 | June 12, 2008 |  |
| 1005 | S.P Sinha | 200,000.00 | 36 | 12.50 | March 1, 2010 |  |
| 1006 | P. Sharma | 700,000.00 | 60 | 12.50 | May 6, 2008 |  |
| 1007 | K.S Dhall | 500,000.00 | 48 | NULL | May 3, 2008 |  |

1. Put the interest rate 11.50% for all the loans for which the interest rate is NULL.
2. Increase the interest rate by 0.5% for all the Loans for which the Loan amount is more than 400000.
3. For each Loan replace Interest with (Loan\_amount \* Int\_rate\* installments)/(12\*100).
4. Delete the records of all the Loans whose start date is before 2008.
5. Delete the records of all the Loans whose name starts with ‘K’
6. Display the details of all the Loans with less than 40 installments.
7. Display the Accno and Loan\_amount of all the loans started before 01-04-2009.
8. Display the int\_rate of all Loans started after 01-04-2009.
9. Display the Accno, cust\_name and Loan amount for all the Loans for which the cust\_name ends with‘Sharma’.
10. Loan\_Amount of all the Loans for which the Cust\_name ends with ‘a’.
11. Display the Accno, Cust\_name and Loan\_Amount for the Loans for which the Cust\_name contains ‘a’.
12. Dsiplay the Accno, Cust\_name and Loan\_Amount for all the Loans for which the Cust\_name does not contain ‘P’.
13. Display the structure of table LOAN\_ACCOUNTS so that you can verify that the table is created as required.
14. Display the details of all the loans in the ascending order of their Loan\_Amount.
15. Display the details of all the Loans in the descending order of their Start\_date.
16. Display the details of all the Loans in the ascending order of their Loan\_amount and within Loan\_amount in the descending order of their Start\_date.
17. Display the Accno, Cust\_name and Loan\_Amount of all the Loans for which the Cust\_name starts with ‘K’.
18. Display the details of all the Loans whose rate of interest in NULL.
19. Display the details of all the loans whose rate of interest is not NULL.
20. Display the amounts of various loans from the table Loan\_Accounts. A Loan\_Amount should appear only once.
21. Display the details of all the loans started after 31-12-2008 for which the number of installments are more than 36.
22. Display the Customer\_name and Loan\_amount for all the Loans for which the Loan amount is less than 500000 or int\_rate is more than 12.
23. Display the details of all Loans which started in the year 2009.
24. Display the details of all the Loans whose Loan amount is in the Range 400000 to 500000.
25. Display the Customer\_name and Loan\_amount of all the Loans for which the number of installments are 26, 36 and 48.
26. Display the customer name, loan\_amount and interest rate. If interest rate is NULL, display it as 0.
27. Display the customer name, loan\_amount and interest rate. If interest rate is NULL, display it as “No Interest”.

**Q.SET 4**

1. Create the table STUDENT\_INFO with Columns: Sid, Stud\_name & stude\_score.

* Insert values into STUDENT\_INFO with the following constraints:Sid should be unique, Stud name NOT NULL and stude\_score DEFAULT value of 20.
* Set Sid as primary key.
* Update stude\_score by adding a value of 5 to stude\_score in the table STUDENT\_INFO for the rows satisfying the condition of stude\_score >150 (Using CASE)

1. Create the tables **worker** and **bonus** with the following fields. The primary key of Worker table is Worker\_ID. Set Worker\_id as foreign key of bonus on update and delete cascade constraints. Each constraint should be given a name

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WORKER\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| 1 | Monika | Arora | 100000 | 2014-02-20 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 | Admin |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 | Admin |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 | Account |
| 7 | Satish | Kumar | 75000 | 2014-01-20 | Account |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 | Admin |

1. **Sample Table – Bonus**

|  |  |  |
| --- | --- | --- |
| **WORKER\_ID** | **BONUS\_DATE** | **BONUS\_AMOUNT** |
| 1 | 2016-02-20 | 5000 |
| 2 | 2016-06-11 | 3000 |
| 3 | 2016-02-20 | 4000 |
| 1 | 2016-02-20 | 4500 |
| 2 | 2016-06-11 | 3500 |

1. Write An SQL Query To Fetch “FIRST\_NAME” From Worker Table Using The Alias Name As <WORKER\_NAME>.
2. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST\_NAME Ascending
3. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.
4. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.
5. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000
6. Write An SQL Query To Fetch “FIRST\_NAME” From Worker Table In Upper Case. (upper())
7. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.
8. Write An SQL Query To Print First Three Characters Of  FIRST\_NAME From Worker Table.( substring())
9. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Removing White Spaces From The Right Side( RTRIM ( ))
10. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side. ( LTRIM ( ))
11. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.( length())
12. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Replacing ‘a’ With ‘A’.( REPLACE( ))
13. Find the First name , last name ,Department, Salary and Bonus of employees whose bonus amount is greater than 4000
14. Delete the employee with worker\_id=7 from worker and display the details of both tables.
15. Drop the foreign key constraint and add a new referential integrity constraint with ‘on update or delete with no action’
16. Delete the employee with worker\_id = 8 from worker.

**Q.SET 5**

**Create the following tables and execute the queries given below**

**SAILORS**

|  |  |  |  |
| --- | --- | --- | --- |
| **sid** | **sname** | **rating** | **age** |
| 22 | Dustin | 7 | 45 |
| 29 | Brutas | 1 | 33 |
| 31 | Lubber | 8 | 55 |
| 32 | Andy | 8 | 25 |
| 58 | Rusty | 10 | 35 |
| 64 | Horatio | 7 | 35 |
| 71 | Zorba | 10 | 16 |
| 74 | Horatio | 9 | 35 |
| 85 | Art | 3 | 26 |
| 95 | Bob | 3 | 64 |

**BOATS**

|  |  |  |
| --- | --- | --- |
| **Bid** | **bname** | **color** |
| 101 | Interlake | Blue |
| 102 | Interlake | Red |
| 103 | Clipper | Green |
| 104 | Marine | Red |

**RESERVES**

|  |  |  |
| --- | --- | --- |
| **sid** | **bid** | **day** |
| 22 | 101 | 10/10/98 |
| 22 | 102 | 10/10/98 |
| 22 | 103 | 10/8/98 |
| 22 | 104 | 10/7/98 |
| 31 | 102 | 11/10/98 |
| 31 | 103 | 11/6/98 |
| 31 | 104 | 11/12/98 |
| 64 | 101 | 9/5/98 |
| 64 | 102 | 9/8/98 |
| 74 | 103 | 9/8/98 |

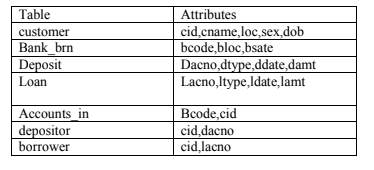
1. Find the names and ages of all sailors
2. Find all information of sailors who have reserved boat number 101.
3. Find all sailors with rating above 7
4. Find the names of sailors who have reserved boat no 103
5. Find the names of sailors who have reserved a red boat, and list in the order of age.
6. Find the names of sailors who have reserved either a red or green boat
7. Find the colors of boats reserved by “Lubber”.
8. Find the names of sailors who have reserved both red and green boats
9. Find the names of sailors who have reserved at least one boat
10. Find the ids and names of sailors who have reserved two different boats on the same day.
11. Find the name and the age of the youngest sailor.
12. Find the names and ratings of sailor whose rating is better than some sailor called Horatio.
13. Find the names of sailors who have reserved all boats.
14. Count the number of different sailor names.
15. Calculate the average age of all sailors.
16. Find the average age of sailors for each rating level.
17. Find the average age of sailors for each rating level that has at least two sailors.

**Q.SET 6**

Consider the database for a banking enterprise. Write the queries for the below

questions.

1. Create the following tables



(ii) Include necessary constraints.

(iii)Tables are created under the database ‘bank’

(iv) Display all the tables in bank database

(v) Describe the structure of all tables

(vi) Delete tables.

***Q.SET 7***

Create the tables given below and execute the queries:

**Customer(Cust id : integer, cust\_name: string)**

**Item(item\_id: integer, item\_name: string, price: integer)**

**Sale(bill\_no: integer, bill\_date: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)**

For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert details of 5 customers, 5 items and 10 sales details. There should be one customer ‘rekha’ who had purchased 3 different products on the same date. And there should be atleast one customer who had purchased 2 different products on the same date in the year ‘2018’.

c) List the details of the customer who have bought a product which has a price>200

d) Give a count of how many products have been bought by each customer group by bill date.

e) Give a count of how many products have been bought by each customer group by bill date only for the year 2018.

f) Give a list of products bought by a customer having cust\_id as 5

g) List the item details which are sold as of today

h) Print the bill in a neat format with the quantity sold, price of the item and the final amount of customer ‘rekha’

**Q.SET 8**

Consider the following database for a banking enterprise.

 BRANCH (bid:int, branch-name: String, branch-city: String, assets: int)

 ACCOUNTS (accno: int, bid:int, balance: int)

 DEPOSITOR (cid:int, accno: int)

 CUSTOMER(cid:int, customer-name:String,customer-street:String,customer-city: String)

Set primary key and foreign keys and insert valid records based on questions.

Write SQL queries to

1. Find all the customers who have at least two accounts at the Main branch.

2. Find all the customers who have an account at all the branches located in a specific city.

3. Find the branch with greatest asset.

4. Find the customer with highest balance.

**Q.SET 9**

Create the tables given below and execute the queries:

Customer(Cust id : integer, cust\_name: string)

Item(item\_id: integer, item\_name: string, price: integer)

Sale(bill\_no: integer, bill\_date: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)

For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert details of 5 customers, 5 items and 10 sales details. There should be one customer

‘rekha’ who had purchased 3 different products on the same date. And there should be

atleast one customer who had purchased 2 different products on the same date in the year

‘2018’.

c) List the details of the customer who have bought a product which has a price&gt;200

d) Give a count of how many products have been bought by each customer group by bill

date.

e) Give a count of how many products have been bought by each customer group by bill

date only for the year 2018.

f) Give a list of products bought by a customer having cust\_id as 5

g) List the item details which are sold as of today

h) Print the bill in a neat format with the quantity sold, price of the item and the final

amount of customer ‘rekha’

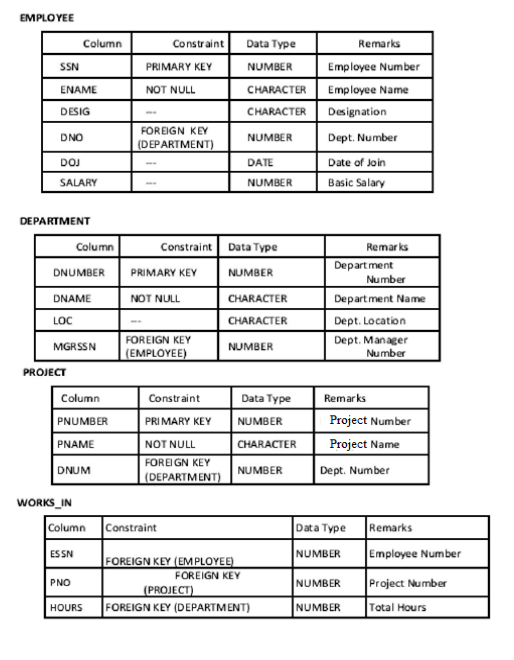
**Q.SET 10**

Create the following tables.

 Primary key, SSN of EMPLOYEE should be created as a sequence starting at 1.

 There should be at least 8 employees and 5 departments

 Check salary range of employees is between 30,000 and 75,000 using check predicate



1. Retrieve all employees in department 5 whose salary is between Rs 30,000 and Rs

40,000.

2. Retrieve a list of employees and the projects they are working on, where the

departments and the employees within the department are alphabetically by name.

3. Retrieve the project number, the project name, and the number of employees who work

in each project.

4. For the project on which more than two employees work, retrieve the project number,

the project name, and the number of employees who work on the project.

5. For each project, retrieve the project number, the project name, and the number of

employees from department 5 who work on the project.

6. For the departments having more than five employees, display the department id and the

number and details of employees earning more than Rs 40,000 per month.

7. Create a synonym for the VIEW created on natural join of emp and dept tables.

8. Use the tables Employee, and Department. Perform the operations as mentioned below:

(a) Display the employee details, departments that the departments are same in

both the emp and dept. (Equi-join)

(b) Display the employee details, departments that the departments are

not same in both the emp and dept. (Non Equi-join)

(c) Perform Left outer join on the emp and dept tables.

(d) Perform Right outer join on the emp and dept tables.

(e) Perform inner join on the emp and dept tables.