# Experiment 1

**TITLE: DDL (Data Definition Language) commands**

**Objective:** To understand the concept of designing issue related to the database with creating, populating the tables.

1. **Create the tables described below: Table name: CLIENT\_MASTER Description:** used to store client information.

|  |  |  |
| --- | --- | --- |
| **Column name** | **data type** | **Size** |
| CLIENTNO | Varchar | 6 |
| NAME | Varchar | 20 |
| ADDRESS 1 | Varchar | 30 |
| ADDRESS 2 | Varchar | 30 |
| CITY | Varchar | 15 |
| PINCODE | Integer |  |
| STATE | Varchar | 15 |
| BALDUE | decimal | 10,2 |

**Table Name: PRODUCT\_MASTER Description:** used to store product information

|  |  |  |
| --- | --- | --- |
| **Column name** | **data type** | **Size** |
| PRODUCTNO | Varchar | 6 |
| DESCRIPTION | Varchar | 15 |
| PROFITPERCENT | Decimal | 4,2 |
| UNIT MEASURE | Varchar | 10 |
| QTYONHAND | Integer |  |

# Table Name:

|  |  |  |
| --- | --- | --- |
| REORDERL VL | Integer |  |
| SELLPRICE | Decimal | 8,2 |
| COSTPRICE | Decimal | 8,2 |

**SALESMAN\_MASTER**

**Description:** Used to store salesman information working for the company.

|  |  |  |
| --- | --- | --- |
| **Column name** | **data type** | **Size** |
| SALESMANNO | Varchar | 6 |
| SALESMANNAME | Varchar | 20 |
| ADDRESS 1 | Varchar | 30 |
| ADDRESS 2 | Varchar | 30 |
| CITY | Varchar | 20 |
| PINCODE | Integer |  |
| STATE | Varchar | 20 |
| SALAMT | Real |  |
| TGTTOGET | Decimal |  |
| YTDSALES | Double | 6,2 |
| REMARKS | Varchar | 60 |

# Insert the following data into their respective tables:

1. Data for **CLIENT\_MASTER** table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Client no | Name | city | Pincode | state | BalDue |
| C00001 | Ivan bayross | Mumbai | 400054 | Maharashtra | 15000 |
| C00002 | Mamta muzumdar | Madras | 780001 | Tamil nadu | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C00003 | Chhaya bankar | Mumbai | 400057 | Maharashtra | 5000 |
| C00004 | Ashwini joshi | Bangalore | 560001 | Karnataka | 0 |
| C00005 | Hansel colaco | Mumbai | 400060 | Maharashtra | 2000 |
| C00006 | Deepak sharma | Mangalore | 560050 | Karnataka | 0 |

1. Data for **PRODUCT**\_**MASTER** table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ProductNo | Description | Profit percent | Unit measure | Qtyonhand | RecorderLvl | SellPrice | CostPrice |
| P00001 | T-Shirt | 5 | Piece | 200 | 50 | 350 | 250 |
| P0345 | Shirts | 6 | Piece | 150 | 50 | 500 | 350 |
| P06734 | Cotton jeans | 5 | Piece | 100 | 20 | 600 | 450 |
| P07865 | Jeans | 5 | Piece | 100 | 20 | 750 | 500 |
| P07868 | Trousers | 2 | Piece | 150 | 50 | 850 | 550 |
| P07885 | Pull Overs | 2.5 | Piece | 80 | 30 | 700 | 450 |
| P07965 | Denim jeans | 4 | Piece | 100 | 40 | 350 | 250 |
| P07975 | Lycra tops | 5 | Piece | 70 | 30 | 300 | 175 |
| P08865 | Skirts | 5 | Piece | 75 | 30 | 450 | 300 |

1. Data for **SALESMAN\_MASTER** table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SalesmanNo** | **Name** | **Address1** | **Address2** | **City** | **PinCode** | **State** |
| S00001 | Aman | A/14 | Worli | Mumbai | 400002 | Maharashtra |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S00002 | Omkar | 65 | Nariman | Mumbai | 400001 | Maharashtra |
| S00003 | Raj | P-7 | Bandra | Mumbai | 400032 | Maharashtra |
| S00004 | Ashish | A/5 | Juhu | Mumbai | 400044 | Maharashtr(a |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SalesmanNo** | **SalAmt** | **TgtToGet** | **YtdSales** | **Remarks** |
| S00001 | 3000 | 100 | 50 | Good |
| S00002 | 3000 | 200 | 100 | Good |
| S00003 | 3000 | 200 | 100 | Good |
| S00004 | 3500 | 200 | 150 | Good |

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Experiment 2

**Title: DML commands with constraints**

**Objective: -** To understand the concept of different DML commands.

Exercise on retrieving records from a table.

1. Find out the names of all the clients.
2. Retrieve the entire contents of the Client\_Master table.
3. Retrieve the list of names,city and the state of all the clients.
4. List the various products available from the Product\_Master table.
5. List all the clients who are located in Mumbai.
6. Find the names of salesman who have a salary equal to Rs.3000.
7. Exercise on updating records in a table
   1. Change the city of ClientNo ‘C00005’ to ‘Bangalore’.
   2. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.
   3. Change the cost price of ‘Trousers’ to rs.950.00.
   4. Change the city of the salesman to Pune.
8. Exercise on deleting records in a table
   1. Delete all salesman from the Salesman\_Master whose salaries are equal to Rs.3500.
9. b. Delete all products from Product\_Master where the quantity on hand is equal to 100.

c. Delete from Client\_Master where the column state holds the value ‘Tamil Nadu’.

1. Exercise on altering the table structure
   1. Add a column called ‘Telephone’ of data type integer to the Client\_Master table.
   2. Change the size off SellPrice column in Product \_Master to 10, 2.
2. Exercise on deleting the table structure along with the data
   1. Destroy the table Client\_Master along with its data.
3. Exercise on renaming the table
   1. Change the name of the Salesman\_Master to sman\_mast.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# EXPERIMENT-3

**TITLE: DDL (Data Definition Language) commands with Data Constraints**

**Objective:** To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key

# Create the tables described below: Table name: CLIENT\_MASTER\_1

**Description:** used to store client information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Constraints** |
| CLIENTNO | Varchar | 6 | Primary key / first letter must start with ‘C’ |
| NAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 |  |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 15 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 15 |  |
| BALDUE | Decimal | 10,2 |  |

**Table Name: PRODUCT\_MASTER\_1 Description:** used to store product information

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| PRODUCTNO | Varchar | 6 | Primary Key/ first letter must start with ‘P’ |
| DESCRIPTION | Varchar | 15 | Not Null |
| PROFITPERCENT | Decimal | 4,2 | Not Null |
| UNIT MEASURE | Varchar | 10 | Not Null |
| QTYONHAND | Integer | 8 | Not Null |
| REORDERL VL | Integer | 8 | Not Null |
| SELLPRICE | Decimal | 8,2 | Not Null |
| COSTPRICE | Decimal | 8,2 | Not Null |

# Table Name: SALESMAN\_MASTER \_1

**Description:** used to store salesman information working for the company.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| SALESMANNO | Varchar | 6 | Primary Key/ first letter must start with ‘S’ |
| SALESMANNAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 | Not Null |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 20 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 20 |  |
| SALAMT | Real | 8,2 | Not Null , Cannot be 0 |
| TGTTOGET | Decimal | 6,2 | Not Null , Cannot be 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| YTDSALES | Double | 6,2 | Not Null |
| REMARKS | Varchar | 60 |  |

# Reinsert the data in these two tables based upon Lab 2.

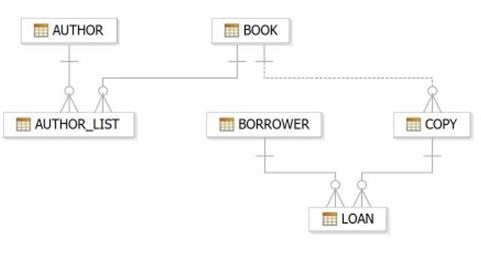
1. **Display the contents of each table.**

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXPERIMENT-4

**TITLE: DDL (Data Definition Language) commands with Data Constraints**

**Objective:** To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key, The Foreign Key and constraints.

Review this diagram



1. Create table AUTHOR = {Author\_ID , Lastname, Firstname, Email, City, Country} Where:

Author\_ID – text data type, 5 characters, primary key Lastname – text data type, 15 characters, not null Firstname – text data type, 15 characters, not null Email – text data type, 40 characters,

City – text data type, 15 characters, Country – text data type, 15 characters,

1. Create Table BOOK={ Book\_ID, Book\_Title, Copies) Where :

Book\_ID – text data type, 5 characters Primary Key Start With Character **B**

Book\_Title - Text data Type Not Null

Copies- No.of copies Data Type int always greater the 2

1. Create table AUTHOR\_LIST = {Author\_ID , Book\_ID , Role} Where:

Author\_ID – text data type, 5 characters, referenced by Author\_ID from AUTHOR table

Book\_ID – text data type, 5 characters Role – text data type, 15 characters

and primary key is: Author\_ID, Book\_ID

1. Add four records in each tables AUTHOR, BOOK, BOOK\_LIST.
2. Alter structure of table AUTHOR\_LIST add the field Publisher data type of 30 Character.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# EXPERIMENT- 5,6

**Title:** Use of Inbuilt functions and relational algebra operation

**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

1. Consider the following table structure and attempt. Supplier-(scode,sname,scity,turnover)

Part-(pcode,weigh,color,cost,sellingprice) Supplier\_Part-(scode,pcode,qty)

1. Create tables
2. Populate the table.
3. Write appropriate SQL Statement for the following:
   1. Get the supplier number and part number in ascending order of supplier number.
   2. Get the details of supplier who operate from Bombay with turnover 50.
   3. Get the total number of supplier.
   4. Get the part number weighing between 25 and 35.
   5. Get the supplier number whose turnover is null.
   6. Get the part number that cost 20, 30 or 40 rupees.
   7. Get the total quantity of part 2 that is supplied.
   8. Get the name of supplier who supply part 2.
   9. Get the part number whose cost is greater than the average cost.
   10. Get the supplier number and turnover in descending order of turnover.

# EXPERIMENT-7,8

**TITLE:** Nested sql queries or Subquries

**Objective:** To understand the use **SQL Subquery**

# Create the following two tables (EMP and DEPT) EMP TABLE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EMPNO DEPTNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM |
|  | | | | | | |
| 7369  20 | SMITH | CLERK | 7902 | 17-DEC-80 | 500 | 800 |
| 7499  30 | ALLEN | SALESMAN | 7698 | 20-FEB-81 | 1600 | 300 |
| 7521  30 | WARD | SALESMAN | 7698 | 22-FEB-81 | 1250 | 500 |
| 7566 JONES MANAGER 7839 02-APR-81 2975 20 | | | | | | |
| 7654  30 | MARTIN | SALESMAN | 7698 | 28-SEP-81 | 1250 | 1400 |
| 7698 BLAKE MANAGER 7839 01-MAY-81 2850 30 | | | | | | |
| 7782 CLARK MANAGER 7839 09-JUN-81 2450 10 | | | | | | |
| 7788 SCOTT ANALYST 7566 09-DEC-82 3000 20 | | | | | | |
| 7839 KING PRESIDENT 17-NOV-81 5000 10 | | | | | | |
| 7844  30 | TURNER | SALESMAN | 7698 | 08-SEP-81 | 1500 | 0 |
| 7876 ADAMS CLERK 7788 12-JAN-83 1100 20 | | | | | | |
| 7900 JAMES CLERK 7698 03-DEC-81 950 30 | | | | | | |
| 7902 FORD ANALYST 7566 03-DEC-81 3000 20 | | | | | | |
| 7934 MILLER CLERK 7782 23-JAN-82 1300 10 | | | | | | |

**DEPT TABLE**

|  |  |  |
| --- | --- | --- |
| DEPTNO | DNAME | LOC |
|  | | |
| 10 | ACCOUNTING | NEW YORK |
| 20 | RESEARCH | DALLAS |
| 30 | SALES | CHICAGO |

|  |
| --- |
| 40 OPERATIONS BOSTON |

Write the Nested Queries for the following queries.

* 1. List the details of the emps whose Salaries more than the employee BLAKE.
  2. List the emps whose Jobs are same as ALLEN.
  3. List the Emps whose Sal is same as FORD or SMITH in desc order of Names.
  4. List the emps Whose Jobs are same as MILLER or Sal is more than ALLEN.
  5. Find the highest paid employee of sales department.
  6. List the employees who are senior to most recently hired employee working under king.

List the employees who are senior to most recently hired employee working under king.

select \* from emp where hiredate < (select max(hiredate) from emp where mgr in (select empno from emp where ename = 'KING')) ;

* 1. List the names of the emps who are getting the highest sal dept wise.

select e.ename,e.deptno from emp e where e.sal in (select max(sal) from emp group by deptno) ;

* 1. List the emps whose sal is equal to the average of max and minimum select \* from emp where sal =(select (max(sal)+min(sal))/2 from emp);
  2. List the emps who joined in the company on the same date.

select \* from emp e where hiredate in (select hiredate from emp where e.empno <> empno);

* 1. Find out the emps who joined in the company before their Managers.

select \* from emp e where hiredate < (select hiredate from emp where empno = e.mgr)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# EXPERIMENT-9

**TITLE: Group by & having clause**

**Objective:** To understand the use of group by and having clause.

Write the SQL Queries for the following queries (use EMP and DEPT table of Exp 8).

1. List the Deptno where there are no emps.
2. List the No.of emp’s and Avg salary within each department for each job.
3. Find the maximum average salary drawn for each job except for ‘President’.
4. List the department details where at least two emps are working.
5. List the no. of emps in each department where the no. is more than 3.
6. List the names of the emps who are getting the highest sal dept wise.
7. List the Deptno and their average salaries for dept with the average salary less than the averages for all departments.

# EXPERIMENT-10

**TITLE: Joins in SQL**

AIM: To execute and verify the SQL commands using Join.

OBJECTIVE: SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.

Refer Experiment 7 & 8 and execute the same questions by using join.