

Create and populate database and perform all data definition language (DDL) commands on your own considered database

-- Create tables

```
CREATE TABLE DEPARTMENT (  
  Dept_ID NUMBER PRIMARY KEY,  
  Dept_Name VARCHAR2(50) NOT NULL,  
  Location VARCHAR2(50)  
);
```

```
CREATE TABLE EMPLOYEE (  
  Emp_ID NUMBER PRIMARY KEY,  
  Emp_Name VARCHAR2(50) NOT NULL,  
  Salary NUMBER CHECK (Salary > 0),  
  Dept_ID NUMBER REFERENCES DEPARTMENT(Dept_ID)  
);
```

```
CREATE TABLE PROJECT (  
  Proj_ID NUMBER PRIMARY KEY,  
  Proj_Name VARCHAR2(50) UNIQUE,  
  Dept_ID NUMBER REFERENCES DEPARTMENT(Dept_ID)  
);
```

-- Insert data

```
INSERT INTO DEPARTMENT VALUES (1, 'HR', 'Pune');  
INSERT INTO DEPARTMENT VALUES (2, 'IT', 'Mumbai');
```

```
INSERT INTO EMPLOYEE VALUES (101, 'Aditya', 50000, 2);  
INSERT INTO EMPLOYEE VALUES (102, 'Riya', 40000, 1);
```

```
INSERT INTO PROJECT VALUES (201, 'Website Dev', 2);  
INSERT INTO PROJECT VALUES (202, 'Recruitment', 1);
```

-- DDL operations

```
ALTER TABLE EMPLOYEE ADD (Email VARCHAR2(50) DEFAULT 'notset@gmail.com');  
ALTER TABLE EMPLOYEE MODIFY (Emp_Name VARCHAR2(100));  
ALTER TABLE PROJECT DROP COLUMN Dept_ID;
```

-- Display data

```
SELECT * FROM DEPARTMENT;  
SELECT * FROM EMPLOYEE;  
SELECT * FROM PROJECT;
```

Perform all data Manipulation language (DML) commands on your own considered database

```
-- Create a table first
CREATE TABLE Employee (
  Emp_ID NUMBER,
  Emp_Name VARCHAR2(50),
  Salary NUMBER,
  City VARCHAR2(50)
);

-- INSERT: Add records into the table
INSERT INTO Employee VALUES (101, 'Aditya', 50000, 'Pune');
INSERT INTO Employee VALUES (102, 'Riya', 45000, 'Mumbai');
INSERT INTO Employee VALUES (103, 'Rahul', 40000, 'Delhi');

-- SELECT: View all data
SELECT * FROM Employee;

-- UPDATE: Modify data in the table
UPDATE Employee SET Salary = 55000 WHERE Emp_ID = 101;

-- DELETE: Remove one record
DELETE FROM Employee WHERE Emp_ID = 103;

-- SELECT again to see changes
SELECT * FROM Employee;
```

Apply Integrity Constraints (NOT NULL, CHECK, UNIQUE, DEFAULT) for your database system

```
CREATE TABLE Student (
  RollNo NUMBER UNIQUE,      -- UNIQUE
  Name VARCHAR2(50) NOT NULL, -- NOT NULL
  Age NUMBER CHECK (Age >= 18), -- CHECK
  City VARCHAR2(50) DEFAULT 'Pune' -- DEFAULT
);

INSERT INTO Student (RollNo, Name, Age) VALUES (1, 'Aditya', 19);
INSERT INTO Student (RollNo, Name, Age, City) VALUES (2, 'Riya', 20, 'Mumbai');

SELECT * FROM Student;
```

Apply foreign key relationship and primary key constraints to relations in your database system.

```
-- Create DEPARTMENT table (Parent table)
CREATE TABLE Department (
  Dept_ID NUMBER PRIMARY KEY,
  Dept_Name VARCHAR2(50)
);

-- Create EMPLOYEE table (Child table)
CREATE TABLE Employee (
  Emp_ID NUMBER PRIMARY KEY,
  Emp_Name VARCHAR2(50),
  Dept_ID NUMBER,
  FOREIGN KEY (Dept_ID) REFERENCES Department(Dept_ID)
);

-- Insert data
INSERT INTO Department VALUES (1, 'HR');
INSERT INTO Department VALUES (2, 'IT');

INSERT INTO Employee VALUES (101, 'Aditya', 2);
INSERT INTO Employee VALUES (102, 'Riya', 1);

-- Display data
SELECT * FROM Department;
SELECT * FROM Employee;
```

Implement and execute Nested queries for your own database system

```
-- Create table Department
CREATE TABLE Department (
  Dept_ID NUMBER PRIMARY KEY,
  Dept_Name VARCHAR2(50)
);

-- Create table Employee
CREATE TABLE Employee (
  Emp_ID NUMBER PRIMARY KEY,
  Emp_Name VARCHAR2(50),
  Salary NUMBER,
  Dept_ID NUMBER REFERENCES Department(Dept_ID)
);

-- Insert data
INSERT INTO Department VALUES (1, 'HR');
INSERT INTO Department VALUES (2, 'IT');
```

```
INSERT INTO Employee VALUES (101, 'Aditya', 50000, 2);
INSERT INTO Employee VALUES (102, 'Riya', 40000, 1);
INSERT INTO Employee VALUES (103, 'Rahul', 60000, 2);
```

```
--①Nested query: find employees who earn more than average salary
SELECT Emp_Name, Salary
FROM Employee
WHERE Salary > (SELECT AVG(Salary) FROM Employee);
```

```
--②Nested query: find employees working in 'IT' department
SELECT Emp_Name
FROM Employee
WHERE Dept_ID = (
    SELECT Dept_ID FROM Department WHERE Dept_Name = 'IT'
);
```

Perform and Implement inner Join operations on your database tables

-- Create Department table

```
CREATE TABLE Department (
    Dept_ID NUMBER PRIMARY KEY,
    Dept_Name VARCHAR2(50)
);
```

-- Create Employee table

```
CREATE TABLE Employee (
    Emp_ID NUMBER PRIMARY KEY,
    Emp_Name VARCHAR2(50),
    Dept_ID NUMBER REFERENCES Department(Dept_ID)
);
```

-- Insert data

```
INSERT INTO Department VALUES (1, 'HR');
INSERT INTO Department VALUES (2, 'IT');
INSERT INTO Department VALUES (3, 'Finance');
```

```
INSERT INTO Employee VALUES (101, 'Aditya', 2);
INSERT INTO Employee VALUES (102, 'Riya', 1);
INSERT INTO Employee VALUES (103, 'Rahul', 2);
```

-- INNER JOIN: shows only matching records

```
SELECT Employee.Emp_ID, Employee.Emp_Name, Department.Dept_Name
FROM Employee
INNER JOIN Department
```

ON Employee.Dept_ID = Department.Dept_ID;

Perform and Implement outer Join operations on your database tables

-- Create Department table

```
CREATE TABLE Department (  
  Dept_ID NUMBER PRIMARY KEY,  
  Dept_Name VARCHAR2(50)  
);
```

-- Create Employee table

```
CREATE TABLE Employee (  
  Emp_ID NUMBER PRIMARY KEY,  
  Emp_Name VARCHAR2(50),  
  Dept_ID NUMBER  
);
```

-- Insert data

```
INSERT INTO Department VALUES (1, 'HR');  
INSERT INTO Department VALUES (2, 'IT');  
INSERT INTO Department VALUES (3, 'Finance');
```

```
INSERT INTO Employee VALUES (101, 'Aditya', 2);  
INSERT INTO Employee VALUES (102, 'Riya', 1);
```

-- Note: No employee in Finance department

-- LEFT OUTER JOIN

```
SELECT e.Emp_Name, d.Dept_Name  
FROM Employee e  
LEFT OUTER JOIN Department d  
ON e.Dept_ID = d.Dept_ID;
```

-- RIGHT OUTER JOIN

```
SELECT e.Emp_Name, d.Dept_Name  
FROM Employee e  
RIGHT OUTER JOIN Department d  
ON e.Dept_ID = d.Dept_ID;
```

Perform and Implement outer Join operations on your database tables

-- Create tables

```
CREATE TABLE Department (  
  Dept_ID NUMBER,  
  Dept_Name VARCHAR2(50)  
);
```

```

CREATE TABLE Employee (
  Emp_ID NUMBER,
  Emp_Name VARCHAR2(50),
  Dept_ID NUMBER
);

-- Insert data
INSERT INTO Department VALUES (1, 'HR');
INSERT INTO Department VALUES (2, 'IT');
INSERT INTO Department VALUES (3, 'Finance');

INSERT INTO Employee VALUES (101, 'Aditya', 2);
INSERT INTO Employee VALUES (102, 'Riya', 1);

-- LEFT OUTER JOIN
SELECT Emp_Name, Dept_Name
FROM Employee
LEFT OUTER JOIN Department
ON Employee.Dept_ID = Department.Dept_ID;

-- RIGHT OUTER JOIN
SELECT Emp_Name, Dept_Name
FROM Employee
RIGHT OUTER JOIN Department
ON Employee.Dept_ID = Department.Dept_ID;

```

Implement views and perform various operations on your database tables

```

-- Create table
CREATE TABLE Employee (
  Emp_ID NUMBER,
  Emp_Name VARCHAR2(50),
  Salary NUMBER,
  Dept VARCHAR2(50)
);

-- Insert data
INSERT INTO Employee VALUES (101, 'Aditya', 50000, 'IT');
INSERT INTO Employee VALUES (102, 'Riya', 45000, 'HR');
INSERT INTO Employee VALUES (103, 'Rahul', 60000, 'IT');

-- Create a VIEW
CREATE VIEW IT_Employees AS
SELECT Emp_Name, Salary
FROM Employee
WHERE Dept = 'IT';

```

```

-- View data from the view
SELECT * FROM IT_Employees;

-- Update through view
UPDATE IT_Employees SET Salary = 65000 WHERE Emp_Name = 'Aditya';

-- Drop the view
DROP VIEW IT_Employees;

```

Implement Trigger on your database tables

```

-- Create main table
CREATE TABLE Employee (
  Emp_ID NUMBER,
  Emp_Name VARCHAR2(50),
  Salary NUMBER
);

-- Create log table to record changes
CREATE TABLE Emp_Log (
  Emp_ID NUMBER,
  Action VARCHAR2(20),
  Action_Time DATE
);

-- Create Trigger
CREATE OR REPLACE TRIGGER emp_after_insert
AFTER INSERT ON Employee
FOR EACH ROW
BEGIN
  INSERT INTO Emp_Log VALUES (:NEW.Emp_ID, 'Inserted', SYSDATE);
END;
/

-- Insert data into main table
INSERT INTO Employee VALUES (101, 'Aditya', 50000);
INSERT INTO Employee VALUES (102, 'Riya', 45000);

-- View both tables
SELECT * FROM Employee;
SELECT * FROM Emp_Log;

```

Create DEPARTMENT Table with columns Department_Id, Name, Location_ID. Constraints on DEPARTMENT table,

Department_Id Primary Key,Location_Id references LOCATION table

```
-- Create LOCATION table (parent table)
CREATE TABLE Location (
  Location_ID NUMBER PRIMARY KEY,
  City VARCHAR2(50)
);

-- Create DEPARTMENT table (child table)
CREATE TABLE Department (
  Department_ID NUMBER PRIMARY KEY,
  Name VARCHAR2(50),
  Location_ID NUMBER REFERENCES Location(Location_ID)
);

-- Insert sample data
INSERT INTO Location VALUES (1, 'Pune');
INSERT INTO Location VALUES (2, 'Mumbai');

INSERT INTO Department VALUES (101, 'HR', 1);
INSERT INTO Department VALUES (102, 'IT', 2);

-- View data
SELECT * FROM Location;
SELECT * FROM Department;
```

Insert the following records into EMPLOYEE table:

EMPL OYEE _ID	LAST_NAME	FIRST_NAM E	MIDDLE _NAME	JOB_ID	MANAG ER_ID	HIRE_DATE	SALARY	COMM	DEPARTM ENT_ID
7369	SMITH	JOHN	Q	667	7902	17-DEC-84	800	NULL	20
7499	ALLEN	KEVIN	J	670	7698	20-FEB-85	1600	300	30
7505	DOYLE	JEAN	K	671	7839	04-APR-85	2850	NULL	30
7506	DENNIS	LYNN	S	671	7839	15-MAY-85	2750	NULL	30
7507	BAKER	LESLIE	D	671	7839	10-JUN-85	2200	NULL	40
7521	WARK	CYNTHIA	D	670	7698	22-FEB-85	1250	500	30

Perform the following queries on the above table

- List all job details.
- List all the locations.
- List out the employee id, name in descending order based on salary

```

-- Create EMPLOYEE table
CREATE TABLE Employee (
  Employee_ID NUMBER,
  Last_Name VARCHAR2(20),
  First_Name VARCHAR2(20),
  Middle_Name VARCHAR2(20),
  Job_ID NUMBER,
  Manager_ID NUMBER,
  Hire_Date DATE,
  Salary NUMBER,
  Comm NUMBER,
  Department_ID NUMBER
);

-- Insert records
INSERT INTO Employee VALUES (7369, 'SMITH', 'JOHN', 'Q', 667, 7902,
TO_DATE('17-DEC-84','DD-MON-YY'), 800, NULL, 20);
INSERT INTO Employee VALUES (7499, 'ALLEN', 'KEVIN', 'J', 670, 7698,
TO_DATE('20-FEB-85','DD-MON-YY'), 1600, 300, 30);
INSERT INTO Employee VALUES (7505, 'DOYLE', 'JEAN', 'K', 671, 7839,
TO_DATE('04-APR-85','DD-MON-YY'), 2850, NULL, 30);
INSERT INTO Employee VALUES (7506, 'DENNIS', 'LYNN', 'S', 671, 7839,
TO_DATE('15-MAY-85','DD-MON-YY'), 2750, NULL, 30);
INSERT INTO Employee VALUES (7507, 'BAKER', 'LESLIE', 'D', 671, 7839,
TO_DATE('10-JUN-85','DD-MON-YY'), 2200, NULL, 40);
INSERT INTO Employee VALUES (7521, 'WARK', 'CYNTHIA', 'D', 670, 7698,
TO_DATE('22-FEB-85','DD-MON-YY'), 1250, 500, 30);

-- View table
SELECT * FROM Employee;

-- (a) List all job details
SELECT DISTINCT Job_ID FROM Employee;

-- (b) List all the locations (Department IDs used as location here)
SELECT DISTINCT Department_ID FROM Employee;

-- (c) List employee id and name in descending order based on salary
SELECT Employee_ID, First_Name, Last_Name, Salary
FROM Employee
ORDER BY Salary DESC;

```

Insert any two records from the below table into EMPLOYEE table:

EMPL OYEE _ID	LAST_NAME	FIRST_NAM E	MIDDLE _NAME	JOB_ID	MANAG ER_ID	HIRE_DATE	SALARY	COMM	DEPARTM ENT_ID
7369	SMITH	JOHN	Q	667	7902	17-DEC-84	800	NULL	20
7499	ALLEN	KEVIN	J	670	7698	20-FEB-85	1600	300	30
7505	DOYLE	JEAN	K	671	7839	04-APR-85	2850	NULL	30

```

-- Create Employee Table
CREATE TABLE Employee (
  Employee_ID NUMBER PRIMARY KEY,
  Last_Name VARCHAR2(20),
  First_Name VARCHAR2(20),
  Middle_Name VARCHAR2(20),
  Job_ID NUMBER,
  Manager_ID NUMBER,
  Hire_Date DATE,
  Salary NUMBER,
  Comm NUMBER,
  Department_ID NUMBER
);

-- Insert sample employee data
INSERT INTO Employee VALUES (7369, 'SMITH', 'JOHN', 'Q', 667, 7902,
TO_DATE('17-DEC-84','DD-MON-YY'), 800, NULL, 20);
INSERT INTO Employee VALUES (7499, 'ALLEN', 'KEVIN', 'J', 670, 7698,
TO_DATE('20-FEB-85','DD-MON-YY'), 1600, 300, 30);
INSERT INTO Employee VALUES (7521, 'WARD', 'ALEX', 'T', 671, 7698,
TO_DATE('22-FEB-85','DD-MON-YY'), 1250, 500, 30);
INSERT INTO Employee VALUES (7566, 'JONES', 'MARTIN', 'L', 672, 7839,
TO_DATE('02-APR-85','DD-MON-YY'), 2975, NULL, 20);
INSERT INTO Employee VALUES (7654, 'MARTIN', 'JAMES', 'B', 673, 7698,
TO_DATE('28-SEP-85','DD-MON-YY'), 1250, 1400, 30);

COMMIT;

-- a) List first name, last name, salary, and commission for all employees
SELECT First_Name, Last_Name, Salary, Comm
FROM Employee;

-- b) List employee's annual salary with their names

```

```
SELECT First_Name, Last_Name, (Salary * 12) AS Annual_Salary
FROM Employee;
```

Insert the following records into EMPLOYEE table:

EMPL OYEE _ID	LAST_NAME	FIRST_NAM E	MIDDLE _NAME	JOB_ID	MANAG ER_ID	HIRE_DATE	SALARY	COMM	DEPARTM ENT_ID
7369	SMITH	JOHN	Q	667	7902	17-DEC-84	800	NULL	20
7499	ALLEN	KEVIN	J	670	7698	20-FEB-85	1600	300	30
7505	DOYLE	JEAN	K	671	7839	04-APR-85	2850	NULL	30
7506	DENNIS	LYNN	S	671	7839	15-MAY-85	2750	NULL	30
7507	BAKER	LESLIE	D	671	7839	10-JUN-85	2200	NULL	40
7521	WARK	CYNTHIA	D	670	7698	22-FEB-85	1250	500	30

Perform the following queries on the above table

- List out the employees who are working in department 20.
- List out the employees who are earning salary between 3000 and 4500.

```
-- Create the EMPLOYEE table
CREATE TABLE Employee (
  Employee_ID NUMBER PRIMARY KEY,
  Last_Name VARCHAR2(20),
  First_Name VARCHAR2(20),
  Middle_Name VARCHAR2(20),
  Job_ID NUMBER,
  Manager_ID NUMBER,
  Hire_Date DATE,
  Salary NUMBER,
  Comm NUMBER,
  Department_ID NUMBER
);
```

-- Insert given records

```
INSERT INTO Employee VALUES (7369, 'SMITH', 'JOHN', 'Q', 667, 7902,
TO_DATE('17-DEC-84','DD-MON-YY'), 800, NULL, 20);
INSERT INTO Employee VALUES (7499, 'ALLEN', 'KEVIN', 'J', 670, 7698,
TO_DATE('20-FEB-85','DD-MON-YY'), 1600, 300, 30);
INSERT INTO Employee VALUES (7505, 'DOYLE', 'JEAN', 'K', 671, 7839,
TO_DATE('04-APR-85','DD-MON-YY'), 2850, NULL, 30);
INSERT INTO Employee VALUES (7506, 'DENNIS', 'LYNN', 'S', 671, 7839,
TO_DATE('15-MAY-85','DD-MON-YY'), 2750, NULL, 30);
INSERT INTO Employee VALUES (7507, 'BAKER', 'LESLIE', 'D', 671, 7839,
TO_DATE('10-JUN-85','DD-MON-YY'), 2200, NULL, 40);
INSERT INTO Employee VALUES (7521, 'WARK', 'CYNTHIA', 'D', 670, 7698,
TO_DATE('22-FEB-85','DD-MON-YY'), 1250, 500, 30);
```

COMMIT;

-- a) Employees working in department 20

```
SELECT First_Name, Last_Name, Department_ID
FROM Employee
WHERE Department_ID = 20;
```

-- b) Employees earning salary between 3000 and 4500

```
SELECT First_Name, Last_Name, Salary
FROM Employee
WHERE Salary BETWEEN 3000 AND 4500;
```

Insert the following records into EMPLOYEE table:

EMPL OYEE _ID	LAST_NAME	FIRST_NAM E	MIDDLE _NAME	JOB_ID	MANAG ER_ID	HIRE_DATE	SALARY	COMM	DEPARTM ENT_ID
7369	SMITH	JOHN	Q	667	7902	17-DEC-84	800	NULL	20
7499	ALLEN	KEVIN	J	670	7698	20-FEB-85	1600	300	30
7505	DOYLE	JEAN	K	671	7839	04-APR-85	2850	NULL	30
7506	DENNIS	LYNN	S	671	7839	15-MAY-85	2750	NULL	30
7507	BAKER	LESLIE	D	671	7839	10-JUN-85	2200	NULL	40
7521	WARK	CYNTHIA	D	670	7698	22-FEB-85	1250	500	30

Perform the following queries on the above table

- List out the employees who are working in department 10 or 20.
- List out the employees whose name starts with "S".
- List out the employees whose name length is 4 and start with "S"

-- Create EMPLOYEE table

```
CREATE TABLE Employee (  
Employee_ID NUMBER PRIMARY KEY,
```

```
Last_Name VARCHAR2(20),
First_Name VARCHAR2(20),
Middle_Name VARCHAR2(20),
Job_ID NUMBER,
Manager_ID NUMBER,
Hire_Date DATE,
Salary NUMBER,
Comm NUMBER,
Department_ID NUMBER
);
```

-- Insert given records

```
INSERT INTO Employee VALUES (7369, 'SMITH', 'JOHN', 'Q', 667, 7902,
TO_DATE('17-DEC-84','DD-MON-YY'), 800, NULL, 20);
INSERT INTO Employee VALUES (7499, 'ALLEN', 'KEVIN', 'J', 670, 7698,
TO_DATE('20-FEB-85','DD-MON-YY'), 1600, 300, 30);
INSERT INTO Employee VALUES (7505, 'DOYLE', 'JEAN', 'K', 671, 7839,
TO_DATE('04-APR-85','DD-MON-YY'), 2850, NULL, 30);
INSERT INTO Employee VALUES (7506, 'DENNIS', 'LYNN', 'S', 671, 7839,
TO_DATE('15-MAY-85','DD-MON-YY'), 2750, NULL, 30);
INSERT INTO Employee VALUES (7507, 'BAKER', 'LESLIE', 'D', 671, 7839,
TO_DATE('10-JUN-85','DD-MON-YY'), 2200, NULL, 40);
INSERT INTO Employee VALUES (7521, 'WARK', 'CYNTHIA', 'D', 670, 7698,
TO_DATE('22-FEB-85','DD-MON-YY'), 1250, 500, 30);
```

COMMIT;

-- a) Employees who are working in department 10 or 20

```
SELECT First_Name, Last_Name, Department_ID
FROM Employee
WHERE Department_ID IN (10, 20);
```

-- b) Employees whose name starts with 'S'

```
SELECT First_Name, Last_Name
FROM Employee
WHERE Last_Name LIKE 'S%';
```

-- c) Employees whose name length is 4 and starts with 'S'

```
SELECT First_Name, Last_Name
FROM Employee
WHERE Last_Name LIKE 'S%' AND LENGTH(Last_Name) = 4;
```