Database Queries and PL/SQL Exercises

```
Database Queries and PL/SQL Exercises

-- 1. Create Database
CREATE DATABASE Banking;

-- Use the created database
USE Banking;

-- 2. Create customer table
CREATE TABLE customer (
cname VARCHAR(50) PRIMARY KEY,
city VARCHAR(50) NOT NULL
);

-- 3. Create branch table
CREATE TABLE branch (
bname VARCHAR(50) PRIMARY KEY,
city VARCHAR(50) NOT NULL CHECK (city IN ('NAGPUR', 'DELHI', 'BANGLORE', 'BOMBAY'))
);

-- 4. Create deposit table
CREATE TABLE deposit (
actno VARCHAR(20) PRIMARY KEY CHECK (actno LIKE 'D%'),
name VARCHAR(50),
bname VARCHAR(50),
amount DECIMAL(8,2) NOT NULL CHECK (amount > 0),
adate DATE,
FOREIGN KEY (name) REFERENCES customer(cname) ON DELETE CASCADE,
FOREIGN KEY (bname) REFERENCES branch(bname) ON DELETE CASCADE
);

-- 5. Create borrow table
CREATE TABLE borrow (
loanno VARCHAR(20) PRIMARY KEY CHECK (loanno LIKE 'L%'),
cname VARCHAR(50),
bname VARCHAR(50),
amount DECIMAL(8,2) NOT NULL CHECK (amount > 0),
FOREIGN KEY (cname) REFERENCES customer(cname) ON DELETE CASCADE,
FOREIGN KEY (bname) REFERENCES branch(bname) ON DELETE CASCADE
);
```

```sql
-- 6. Retrieve information from customer table
SELECT * FROM customer;

-- 7. Retrieve information from branch table
SELECT * FROM branch;

-- 8. Create a view for customers residing in Delhi
CREATE VIEW CustomersInDelhi AS
SELECT * FROM customer WHERE city = 'DELHI';

-- 9. Retrieve information from deposit table
SELECT * FROM deposit;

-- 10. Create a view for deposits greater than Rs 5000
CREATE VIEW HighValueDeposits AS
SELECT * FROM deposit WHERE amount > 5000;

-- 11. Create emp table
CREATE TABLE emp (
empno INT(4) PRIMARY KEY,
ename CHAR(10),
hiredate DATE,
salary DECIMAL(10,2),
commission DECIMAL(10,2)
);

-- 12. Insert sample data
INSERT INTO emp (empno, ename, hiredate, salary, commission) VALUES
(101, 'John', '2023-01-15', 2000.00, 500.00),
(102, 'Alice', '2022-11-20', 2500.00, 700.00),
(103, 'Bob', '2023-03-10', 1800.00, 400.00),
(104, 'David', '2021-07-05', 3000.00, 900.00),
(105, 'Emma', '2020-12-25', 2800.00, 800.00);

-- 13. Add Check Constraint (Salary >= 1500)
ALTER TABLE emp ADD CONSTRAINT chk_salary CHECK (salary >= 1500);

-- 14. Add Check Constraint (empno length <= 3)
ALTER TABLE emp ADD CONSTRAINT chk_empno CHECK (LENGTH(empno) <= 3);

-- 15. Add new column dept
ALTER TABLE emp ADD COLUMN dept VARCHAR(5);

-- 16. Rename column dept to department
ALTER TABLE emp CHANGE dept department VARCHAR(5);
```

```sql
-- 17. Modify datatype of department
ALTER TABLE emp MODIFY department VARCHAR(10);

-- 18. Rename table emp to employee
ALTER TABLE emp RENAME TO employee;

-- 19. Truncate table employee
TRUNCATE TABLE employee;

-- 20. Drop table employee
DROP TABLE employee;

-- 21. Show tables
SHOW TABLES;

-- 22. Aggregate functions

-- List total loan
SELECT SUM(amount) AS total_loan FROM borrow;

-- List total deposit
SELECT SUM(amount) AS total_deposit FROM deposit;

-- List total loan taken from 'NAGPUR' branch
SELECT SUM(amount) AS total_loan_nagpur FROM borrow WHERE bname = 'NAGPUR';

-- List total deposit of customers having account date later than 1-Jan-96
SELECT SUM(amount) AS total_deposit_after_1996 FROM deposit WHERE adate > '1996-01-01';

-- List total deposit of customers living in city NAGPUR
SELECT SUM(d.amount) AS total_deposit_nagpur FROM deposit d JOIN customer c ON d.name = c.cname
WHERE c.city = 'NAGPUR';

-- List maximum deposit of customer living in BOMBAY
SELECT MAX(d.amount) AS max_deposit_bombay FROM deposit d JOIN customer c ON d.name = c.cname
WHERE c.city = 'BOMBAY';

-- List total deposit of customers having branch in BOMBAY
SELECT SUM(amount) AS total_deposit_bombay FROM deposit WHERE bname = 'BOMBAY';

-- Count total number of branch cities
SELECT COUNT(DISTINCT city) AS total_branch_cities FROM branch;

-- Count total number of customer cities
```

```
SELECT COUNT(DISTINCT city) AS total_customer_cities FROM customer;

-- Group By and Having Clause

-- List the branches having a sum of deposit more than 50000
SELECT bname, SUM(amount) AS total_deposit FROM deposit GROUP BY bname HAVING SUM(amount) > 50000;

-- List the branches having sum of deposit more than 5000 and located in city BOMBAY
SELECT bname, SUM(amount) AS total_deposit FROM deposit JOIN branch USING (bname) WHERE city = 'BOMBAY' GROUP BY bname HAVING SUM(amount) > 5000;

-- List the names of customers having maximum deposit
SELECT name FROM deposit WHERE amount = (SELECT MAX(amount) FROM deposit);

-- List the names of customers having deposited in the branches where the average deposit is more than 5000
SELECT DISTINCT name FROM deposit WHERE bname IN (SELECT bname FROM deposit GROUP BY bname HAVING AVG(amount) > 5000);

-- List the name of the branch having the highest number of depositors
SELECT bname FROM deposit GROUP BY bname ORDER BY COUNT(name) DESC LIMIT 1;

-- Count the number of depositors living in KOCHI
SELECT COUNT(DISTINCT name) FROM deposit JOIN customer USING (name) WHERE city = 'KOCHI';

-- Give the names of cities in which the maximum number of branches are located
SELECT city FROM branch GROUP BY city ORDER BY COUNT(bname) DESC LIMIT 1;

-- Count the number of customers living in the city where branch is located
SELECT COUNT(DISTINCT cname) FROM customer JOIN branch USING (city);

-- PL/SQL block to accept two numbers and print their product
DECLARE
num1 NUMBER;
num2 NUMBER;
product NUMBER;
BEGIN
num1 := &Enter_Number1;
num2 := &Enter_Number2;
product := num1 * num2;
DBMS_OUTPUT.PUT_LINE('The product is: ' || product);
END; /
```