# Faiz Khan

**Assignment 3**

```
DROP TABLE MEMBER PURGE;
DROP TABLE BOOKS PURGE;
DROP TABLE ISSUE PURGE;

--1)Create the table Member, Books and Issue without any constraints as mentioned
in the schema description above.
CREATE TABLE Member (
    Member_Id NUMBER(5),
    Member_Name VARCHAR2(30),
    Member_Address VARCHAR2(50),
    Acc_Open_Date DATE,
    Membership_type VARCHAR2(20),
    Fees_Paid NUMBER(4),
    Max_Books_Allowed NUMBER(2),
    Penalty_Amount NUMBER(7,2)
);

CREATE TABLE Books (
    Book_No NUMBER(6),
    Book_Name VARCHAR2(30),
    Author_Name VARCHAR2(30),
    Cost NUMBER(7,2),
    Category CHAR(10)
);

CREATE TABLE Issue (
    Lib_Issue_Id NUMBER(10),
    Book_No NUMBER(6),
    Member_Id NUMBER(5),
    Issue_Date DATE,
    Return_Date DATE
);


--2)View the structure of the tables.
DESC Member;
DESC Books;
DESC Issue;

--3)Drop the Member table
DROP TABLE Member;

--4)Create the table Member again as per the schema description with the following
constraints.
--a.    Member_Id – Primary Key
--b.    Membership_type - ‘Lifetime’,’ Annual’, ‘Half Yearly’,’ Quarterly’
CREATE TABLE Member (
    Member_Id NUMBER(5) PRIMARY KEY,
    Member_Name VARCHAR2(30),
    Member_Address VARCHAR2(50),
    Acc_Open_Date DATE,
    Membership_type VARCHAR2(20) CHECK (Membership_type IN
('Lifetime','Annual','Half Yearly','Quarterly')),
    Fees_Paid NUMBER(4),
```

```sql
    Max_Books_Allowed NUMBER(2),
    Penalty_Amount NUMBER(7,2)
);

--5)Modify the table Member increase the width of the member name to 30 characters.
ALTER TABLE Member MODIFY (Member_Name VARCHAR2(30));

--6)Add a column named as Reference of Char(30) to Issue table.
ALTER TABLE Issue ADD Reference CHAR(30);

-- 7) Drop the column Reference from Issue
ALTER TABLE Issue DROP COLUMN REFERENCE;

--8)Rename the table Issue to Lib_Issue.
RENAME Issue to Lib_Issue;

--9)Insert following data in table Member
INSERT INTO MEMBER VALUES (1, 'Richa Sharma', 'Pune', TO_DATE('10-12-2005','DD-MM-
YYYY'), 'Lifetime', 25000, 5, 50);
INSERT INTO MEMBER VALUES (2, 'Garima Sen', 'Pune', SYSDATE, 'Annual', 1000, 3,
NULL);
SELECT * FROM MEMBER
--Does not allow fees_paid 25000 outside precision


--10)Insert at least 5 records with suitable data and save it.
INSERT INTO Member VALUES (3, 'Amit Patel', 'Mumbai', SYSDATE, 'Quarterly', 500, 2,
NULL);
INSERT INTO Member VALUES (4, 'Suresh Rao', 'Delhi', SYSDATE, 'Half Yearly', 2000,
4, 0);
INSERT INTO Member VALUES (5, 'Neha Gupta', 'Pune', SYSDATE, 'Annual', 1200, 3, 0);
INSERT INTO Member VALUES (6, 'Rohan Das', 'Chennai', SYSDATE, 'Lifetime', 30000,
6, NULL);
INSERT INTO Member VALUES (7, 'Priya Nair', 'Kolkata', SYSDATE, 'Annual', 1500, 3,
0);


--11)Modify the column Member_name. Decrease the width of the member  name to 20
characters. (If it does not allow state the reason for that)
ALTER TABLE Member MODIFY (Member_Name VARCHAR2(20));

--12)Try to insert a record with Max_Books_Allowed = 110, Observe   the error that
comes. Report the reason for this error.
INSERT INTO Member VALUES (8, 'Test User', 'Test City', SYSDATE, 'Annual', 500,
110, 0);
--Outside precision


--13)Generate another table named Member101 using a Create command along with a
simple SQL query on member table.
CREATE TABLE Member101 AS SELECT * FROM Member;
SELECT * FROM Member101;

--14)   Add the constraints on columns max_books_allowed  and penalty_amt as
follows
--a.    max_books_allowed  < 100
--b.    penalty_amt maximum 1000
    -- Also give names to the constraints
```

```
ALTER TABLE Member ADD CONSTRAINT chk_max_books CHECK (Max_Books_Allowed < 100);
ALTER TABLE Member ADD CONSTRAINT chk_penalty CHECK (Penalty_Amount <= 1000);

--15)Drop the table books
DROP TABLE BOOKS;

--16)   Create table Books again as per the schema description with the following
constraints.
    --a.    Book_No – Primary Key
    --b.    Book_Name – Not Null
    --c.    Category – Science, Fiction, Database, RDBMS, Others.

CREATE TABLE Books (
    Book_No NUMBER(6) PRIMARY KEY,
    Book_Name VARCHAR2(30) NOT NULL,
    Author_Name VARCHAR2(30),
    Cost NUMBER(7,2),
    Category VARCHAR2(20),
    CHECK (Category IN ('Science','Fiction','Database','RDBMS','Others'))
);

--17)Insert data in Book table as follows:
INSERT INTO Books VALUES (101, 'Let us C', 'Denis Ritchie', 450, 'Others');
INSERT INTO Books VALUES (102, 'Oracle – Complete Ref', 'Loni', 550, 'Database');
INSERT       INTO Books VALUES (103, 'Mastering SQL', 'Loni', 250, 'Database');
INSERT INTO Books VALUES (104, 'PL SQL-Ref', 'Scott Urman', 750, 'Database');
SELECT * FROM BOOKS;

--18)Insert more records in Book table using & operator in the insert statement.
INSERT INTO BOOKS VALUES (&Bookid, '&bookname', '&authorname', &authorid,
'&genre');
--& is a placeholder which prompts user to enter value

--19)Create table Book101 similar to Book in structure with no data in it.
CREATE TABLE Book101 AS SELECT * FROM Books WHERE False;
SELECT * FROM Book101

--20)Insert into Book101 all the data in Book table using Select Statement.
INSERT INTO Book101 (SELECT * FROM Books);

--21)Save all the data so far inserted in the tables.
COMMIT;

--22)View the data in the tables using simple SQL query.
SELECT * FROM Member;
SELECT * FROM Books;
SELECT * FROM Book101;
SELECT * FROM Lib_Issue;

--23)   Insert into Book following data. 106,  National Geographic, Adis Scott,
1000,  Science
INSERT INTO Books VALUES (106, 'National Geographic', 'Adis Scott', 1000,
'Science');

--24)    Undo the last changes.
ROLLBACK;
SELECT * FROM Books;
```

```
--25)   Modify the price of book with id 103 to Rs 300 and category to RDBMS.
UPDATE Books SET Cost = 300, Category = 'RDBMS' WHERE Book_No = 103;

--26)   Rename the table Lib_Issue to Issue.
RENAME Lib_Issue to Issue;

--27)   Drop table Issue.
DROP TABLE Issue

--28)   As per the given structure Create table Issue again with following
constraints.
-- Lib_Issue_Id-Primary key
-- Book_No-  foreign key
-- Member_id - foreign key
-- Issue_date < Return_date

CREATE TABLE Issue (
    Lib_Issue_Id NUMBER(10) PRIMARY KEY,
    Book_No NUMBER(6),
    Member_Id NUMBER(5),
    Issue_Date DATE,
    Return_Date DATE,
    CONSTRAINT fk_book FOREIGN KEY (Book_No) REFERENCES Books(Book_No),
    CONSTRAINT fk_member FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id),
    CONSTRAINT chk_dates CHECK (Return_Date IS NULL OR Issue_Date < Return_Date)
);

--29) Insert following data into Issue table
INSERT INTO Issue VALUES(7001, 101, 1, '1-Dec-2006', Null);

--30)   Save the data.
COMMIT;


-- 31)

SELECT M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE FROM MEMBER M, ISSUE I

WHERE M.MEMBER_ID = I.MEMBER_ID AND I.ISSUE_DATE BETWEEN

    TO_DATE('01-DEC-2025','DD-MON-YYYY')

    AND TO_DATE('31-DEC-2025','DD-MON-YYYY');



-- 32)

SELECT DISTINCT M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE FROM MEMBER M, ISSUE
I

WHERE M.MEMBER_ID = I.MEMBER_ID AND I.RETURN_DATE IS NULL;
```

```sql
-- 33)

SELECT M2.MEMBER_ID, M2.MEMBER_NAME, M2.MEMBERSHIP_TYPE, M2.ACC_OPEN_DATE FROM
MEMBER M1, MEMBER M2

WHERE M1.MEMBER_NAME LIKE '%GARIMA%' AND M1.ACC_OPEN_DATE = M2.ACC_OPEN_DATE;




-- 34)

SELECT M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE FROM MEMBER M, ISSUE I, BOOKS
B

WHERE M.MEMBER_ID = I.MEMBER_ID

    AND I.BOOK_NO = B.BOOK_NO

    AND LOWER(B.AUTHOR_NAME) = '%loni%'

    AND TO_CHAR(I.ISSUE_DATE, 'Mon') = 'Dec';


-- 35)

SELECT B.AUTHOR_NAME, COUNT(*) AS TOTAL_ISSUES

FROM MEMBER M, ISSUE I, BOOKS B

WHERE M.MEMBER_ID = I.MEMBER_ID

  AND I.BOOK_NO = B.BOOK_NO

  AND M.MEMBERSHIP_TYPE = 'Lifetime'

GROUP BY B.AUTHOR_NAME

HAVING COUNT(*) = (

    SELECT MIN(CNT)

    FROM (

        SELECT COUNT(*) AS CNT

        FROM MEMBER M2, ISSUE I2, BOOKS B2

        WHERE M2.MEMBER_ID = I2.MEMBER_ID

          AND I2.BOOK_NO = B2.BOOK_NO

          AND M2.MEMBERSHIP_TYPE = 'Lifetime'

        GROUP BY B2.AUTHOR_NAME
    )

);



-- 36)

SELECT AUTHOR_NAME, TOTAL_ISSUES

FROM (

    SELECT B.AUTHOR_NAME, COUNT(*) AS TOTAL_ISSUES

    FROM MEMBER M, ISSUE I, BOOKS B

    WHERE M.MEMBER_ID = I.MEMBER_ID

      AND I.BOOK_NO = B.BOOK_NO

      AND M.MEMBERSHIP_TYPE = 'Half Yearly'

    GROUP BY B.AUTHOR_NAME

    ORDER BY TOTAL_ISSUES DESC

)

WHERE ROWNUM <= 3;



-- 37)

SELECT BOOK_NO, BOOK_NAME, AUTHOR_NAME, TOTAL_ISSUES

FROM (

    SELECT B.BOOK_NO, B.BOOK_NAME, B.AUTHOR_NAME, COUNT(*) AS TOTAL_ISSUES

    FROM MEMBER M, ISSUE I, BOOKS B

    WHERE M.MEMBER_ID = I.MEMBER_ID

      AND I.BOOK_NO = B.BOOK_NO

      AND M.MEMBERSHIP_TYPE = 'Annual'

    GROUP BY B.BOOK_NO, B.BOOK_NAME, B.AUTHOR_NAME

    ORDER BY TOTAL_ISSUES DESC

)
```

```
WHERE ROWNUM <= 5;



-- 38)

SELECT M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE

FROM MEMBER M, ISSUE I, BOOKS B

WHERE M.MEMBER_ID = I.MEMBER_ID

  AND I.BOOK_NO = B.BOOK_NO

  AND B.COST > 300

  AND B.AUTHOR_NAME = 'Scott Urman';



-- 39)

SELECT M.MEMBERSHIP_TYPE AS "MEMBER TYPE",

       B.CATEGORY AS "BOOK CATEGORY",

       COUNT(*) AS "COUNT"

FROM MEMBER M, ISSUE I, BOOKS B

WHERE M.MEMBER_ID = I.MEMBER_ID

  AND I.BOOK_NO = B.BOOK_NO

GROUP BY M.MEMBERSHIP_TYPE, B.CATEGORY

ORDER BY M.MEMBERSHIP_TYPE, B.CATEGORY;



-- 40)

SELECT M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE, M.ACC_OPEN_DATE

FROM MEMBER M, ISSUE I

WHERE M.MEMBER_ID = I.MEMBER_ID

  AND M.MEMBERSHIP_TYPE = 'Lifetime'

  AND M.ACC_OPEN_DATE BETWEEN TO_DATE('01-JAN-2006','DD-MON-YYYY')
```

```
                          AND TO_DATE('31-DEC-2006','DD-MON-YYYY')
GROUP BY M.MEMBER_ID, M.MEMBER_NAME, M.MEMBERSHIP_TYPE, M.ACC_OPEN_DATE
HAVING COUNT(I.BOOK_NO) = 1;
```